Universidade Federal de Pernambuco
Centro de Tecnologia e Geociências
Curso de Especialização em Engenharia de Instrumentação Industrial

Supervisão da medição de grandezas em uma planta industrial via internet

Fabiano Gabriel da Silva

Orientador: Prof. Edval J. P. Santos, PhD

Monografia apresentada ao Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco como parte dos requisitos para obtenção do Certificado de Especialista em Engenharia de Instrumentação Industrial

Recife, 2008

Resumo

Supervisão da medição de grandezas em uma planta industrial via internet

Fabiano Gabriel da Silva

Dezembro/2008

Orientador: Prof. Edval J. P. Santos, PhD
Área de concentração: Engenharia Elétrica
Palavras-chaves: Instrumentação eletrônica, Automação industrial, Redes, CLP, *WebGate*, internet, intranet , *HTML*

A tecnologia da internet, protocolo TCP/IP, oferece uma solução de baixo custo para implementação de sistemas supervisórios. Esse trabalho consiste em avaliar a tecnologia de interface com a internet para controladores lógicos programáveis, CLP, fabricados pela ALTUS SA. Para essa avaliação foi utilizada uma planta industrial existente no laboratório, a qual utiliza CLP da série ponto da ALTUS.

A comunicação entre o CLP e a rede foi feita através do *Gateway* da Altus (*WebGate*). As consultas às informações da planta industrial foram feitas utilizando páginas *HTML* (*Hyper Text Markup Language)* com recursos de *Applets Java* e *Java Script*.

O *Applet* utilizado no trabalho faz parte do *WebView* que é um sistema composto por um banco de *applets* utilizados para desenvolver páginas *HTML*. Utilizamos o *Applet Penchart* para elaboração do gráfico de variação de grandezas (tensão, corrente, velocidade) num determinado período de tempo.

A linguagem *Java Script* foi utilizada para leitura e escrita de operandos. Ela permite o envio de comandos ao *WebGate* que são respondidos em linguagem XML formatada por estilos (arquivos .xsl) que ficam dentro do *WebGate*. As páginas desenvolvidas possibilitam a supervisão em tempo real dos resultados do processo.

LISTA DE FIGURAS

FIGURA 1.1 – MODELO DE PLANTA INDUSTRIAL - VISTA LATERAL 11
FIGURA 1.2 - LIGAÇÃO DO *WEBGATE* COM A INTERNET ATRAVÉS DE *FIREWALL* 14
FIGURA 1.3 – SISTEMA DE COMUNICAÇÃO COM *FIREWALL E WEBGATE* 14
FIGURA 2.1- ELEMENTOS DA SÉRIE PONTO 18
FIGURA 2.2 - MATRIZ DE PROGRAMAÇÃO 21
FIGURA 2.3 - TELA DE MÓDULOS DE CRIAÇÃO DO PROJETO 23
FIGURA 2.4 – MODOS DE OPERAÇÃO DO CLP 24
FIGURA 2.5 – PROCESSO DE EXECUÇÃO DE UM PROGRAMA 25
FIGURA 2.6 - MÓDULO INVERSOR WEG CFW09 27
FIGURA 2.7 – IHM CIMREX 12 27
FIGURA 2.8 – DIAGRAMA DE ALIMENTAÇÃO DO *WEBGATE* PO9900 28
FIGURA 2.9 – EXEMPLO DE ACESSO VIA REDE EXTERNA E INTERNA 33
FIGURA 2.10 – EXEMPLO DE ACESSO AO CLP VIA PROTOCOLO ALNET I 33
FIGURA 2.11 – INTERFACE DE ACESSO AO *WEBGATE* 34
FIGURA 3.1 - NOME DA CONEXÃO 41
FIGURA 3.2 – PORTA QUE SERÁ ACESSADA 42
FIGURA 3.3 – TELA DE PARÂMETROS DE CONFIGURAÇÃO DO HIPER TERMINAL 42
FIGURA 3.4 - TELA DE INICIALIZAÇÃO DO *WEBGATE* VIA HYPER TERMINAL 43
FIGURA 3.5 – TELA DE VISUALIZAÇÃO DO DIRETÓRIO DO *WEBGATE* 43
FIGURA 3.6- CONEXÃO ENTRE *WEBGATE* – PC E *WEBGATE* – CLP PO3042 44
FIGURA 3.7 - BANCADA DE SIMULAÇÕES E TESTES COM CLP PO3042 44
FIGURA 3.8 - CONEXÃO ENTRE *WEBGATE*–PC E *WEBGATE*–CLP PO3042–CIMREX 12 45
FIGURA 3.9– DIAGRAMA DA VISTA FRONTAL DA BANCADA VISTA 45
FIGURA 3.10 - PÁGINA HTML PARA SUPERVISÃO DA PLANTA INDUSTRIAL 46
FIGURA 3.11 – PLANTA INDUSTRIAL COM EQUIPAMENTOS MONTADOS 46
FIGURA 3.12 – TELA DE ACESSO AO WS_FTP 47
FIGURA 3.13 – TELA DE TRANSFERÊNCIA DE ARQUIVOS DO WS_FTP 47
FIGURA 3.14 - TESTE DE COMUNICAÇÃO VIA BROWSER 48
FIGURA 3.15 – ALTERAÇÃO DO VALOR DE VARIÁVEL COM COMANDO 16 49
FIGURA 3.16 – CONFIGURAÇAO DO CANAL ETHERNET 49
FIGURA 3.17 – MONITORAMENTO DAS VARIÁVEIS %S0050.1 E %S0050.2 50
FIGURA 3.18 - ESTRUTURA UTILIZADA NA SIMULAÇÃO 51
FIGURA 3.19 - FONTE DE TENSÃO E CORRENTE 51
FIGURA 3.20 - GERADOR DE SINAL 51
FIGURA 3.21 – CONFIGURAÇÃO DE CANAIS DO PO1112 52
FIGURA 3.22 – DIAGRAMA DE LIGAÇÃO DE TENSÃO 52
FIGURA 3.23 – DIAGRAMA DE LIGAÇÃO DE FONTE DE CORRENTE 52
FIGURA 3.24 – DIAGRAMA DE LIGAÇÃO DE RESISTOR E TERMISTOR 53
FIGURA 3.25 – ONDA SENOIDAL COM FREQUÊNCIA DE 20HZ – 16VPP 53
FIGURA 3.28 – FORMAS DE ONDA SENOIDAL, QUADRADA TRIANGULAR PARA 50HZ 16VPP. 54
FIGURA 3.29 - FORMAS DE ONDA SENOIDAL, QUADRADA TRIANGULAR PARA 100HZ 16VPP 54
FIGURA 3.30 - CONFIGURAÇÃO DA TABELAS 55
FIGURA 3.31 – MONITORAMENTO DE OPERANDOS TABELA 55

FIGURA 3.32 – LÓGICAS PARA GRAVAÇÃO E LEITURA DE DADOS NUMA TABELA 57
FIGURA A.1– TELA INICIAL DE CONFIGURAÇÃO DO BARRAMENTO 60
FIGURA A.2 - TELA DE CONFIGURAÇÃO DOS SEGMENTOS 60
FIGURA A.3 – TELA COM ESPECIFICAÇÕES DO BARRAMENTO 61
FIGURA A.4 – TELA DO PROPONTO 61
FIGURA A.5 - PARÂMETROS DE CONFIGURAÇÃO DA BASE DE ENTRADA PO1112 62
FIGURA A.6 - PARÂMETROS DE CONFIGURAÇÃO DA BASE DE SAÍDA PO2132 62
FIGURA A.7 - TELA PARA SALVAR AS CONFIGURAÇÕES FEITAS NO PROPONTO 62
FIGURA A.8 – DIAGRAMA DE LIGAÇÕES ELÉTRICAS DO PO1112 68

LISTA DE TABELAS

TABELA 2.1– MODELOS DA ARQUITETURA PONTO 18
TABELA 2.2 – TIPOS DISPONÍVEIS DE UCP 20
TABELA 2.3 – LED´S DE DIAGNÓSTICO DA UCP PO3045 26
TABELA 2.4 – MODOS DE EXECUÇÃO DA UCP 26
TABELA 2.5 – CABOS DE CONEXÃO 29
TABELA 2.6 – TIPOS DE PROTOCOLO 29
TABELA 2.7 COMPARAÇÃO ENTRE OS MODELOS OSI, TCP/IP E ALTUS 30
TABELA 2.8 – NÚMERO MÁXIMO DE OPERANDOS POR BLOCO DE DADOS 32
TABELA 2.9 - ESTRUTURA DE UMA PÁGINA *HTML* COM XML E JAVA SCRIPT 35
TABELA 2.10 – ESCRITA DE OPERANDOS EM JAVA SCRIPT 36
TABELA 2.11 –LEITURA DE OPERANDOS EM JAVA SCRIPT 36
TABELA 2.12– *APPLETS* JAVA DISPONÍVEIS PARA IMPLANTAÇÃO DE APLICAÇÕES 38
TABELA 2.13 – PARÂMETROS DE *APPLETS* JAVA 38
TABELA 2.14– LIMITES DO *WEBVIEW* 39
TABELA 2.15 – DESEMPENHO 40
TABELA 3.1 - CÓDIGOS DE COMANDO 48
TABELA 3.2 – TABELA COM PACOTES ENVIADOS E RECEBIDOS 50
TABELA A.1 – CRONOGRAMA DE DESENVOLVIMENTO DA MONOGRAFIA 67
TABELA A.2 - CARACTERÍSTICAS DO MODO DE TENSÃO 69
TABELA A.3 – CARACTERÍSTICAS DO MODO DE CORRENTE 69
TABELA A.4 – CARACTERÍSTICAS DO MODE DE RESISTÊNCIA 70

LISTA DE SIGLAS

WWW	Word Wide Web
HTML	Hyper Text Markup Language
HTTP	HyperText Transfer Protocol
FTP	File Transfer Protocol
URL	Universal Resource Locator
IP	Internet Protocol
TCP	Transmisson Control Protoco
CLP	Controlador Lógico Programável
PLC	Programmable Logic Controller
CP	Controlador Programável
LD	Ladder Diagram
UCP	Unit Central Processing
RAM	Random Access Memory
EPROM	Electrically Programmable Read Only Memory
XML	Extensible Markup Language
XSL	Extensible Style Language
DHCP	Dynamic Host Configuration Protocol
LDN	Laboratório de Dispositivos e Nanoestruturas
cc	Corrente contínua
ca	Corrente alternada
Site	Página da internet
IHM	Interface homem máquina
ST	Linguagem estruturada
OSI	Open Systems Interconection
ISO	Internation Standard Organization
MAC	Media Acess Control
LAN	Local Area Network
WAN	Wide Area Network

SUMÁRIO

CAPÍTULO 1 INTRODUÇÃO 9

1.1 MOTIVAÇÃO 9
1.2 SITUAÇÃO ANTERIOR 12
1.3 METODOLOGIA 12
1.4 SEGURANÇA 13

CAPÍTULO 2 TECNOLOGIAS ENVOLVIDAS 15

2.1 CONTROLADOR LÓGICO PROGRAMÁVEL 15
2.1.1 Histórico 15
2.1.2 Principio de funcionamento 16
2.1.3 Componentes básicos 16
2.1.4 Vantagens 17
2.1.5 Desvantagens 17
2.1.6 Elementos da arquitetura ponto 17
2.2 INVERSOR DE FREQUÊNCIA CFW09 27
2.3 INTERFACE IHM CIMREX 12 27
2.4 ETHERNET 27
2.5 WEBGATE 28
2.5.1 Características da interface PO9900 28
2.5.2 Alimentação 28
2.5.3 Servidor web 28
2.5.4 Acesso ao WebGate 29
2.5.5 Conexão do WebGate com o CLP 29
2.5.6 Conexão com a rede ethernet 29
2.6 PROTOCOLOS 29
2.6.1 ALNET 30
2.6.2 TCP/IP 34
2.6.3 HTTP 34
2.6.4 FTP 35
2.7 CRIAÇÃO DE PÁGINAS HTML 35
2.8 WEBVIEW 37

CAPÍTULO 3 ANÁLISE E TESTES DE DESEMPENHO 41

CAPÍTULO 4 CONCLUSÕES E TRABALHOS FUTUROS 58

APÊNDICE A 60

A-1 CONFIGURANDO PROPONTO 60
A-2 CÓDIGO FONTE DA PÁGINA HTML 63
A-3 CÓDIGO FONTE DE ESTILO 66
A-4 CÓDIGO FONTE DA PÁGINA COM GRÁFICO PENCHART 67
A-5 CRONOGRAMA 67
A-6 DIAGRAMA DE LIGAÇÕES ELÉTRICAS PO1112 68

A-7 MODOS DE TENSÃO, CORRENTE E RESISTÊNCIA. 69
REFERÊNCIAS BIBLIOGRÁFICAS..... 72

CAPÍTULO 1

INTRODUÇÃO

A comunicação à longa distância é uma coisa necessária no cotidiano das pessoas. Ela não é de hoje nem de ontem, começou com o invento do telefone.

Nos dias atuais vivemos uma nova revolução na comunicação a longa distância: expansão da telefonia móvel, popularização da internet e a convergência de redes que seria a união das mais variadas formas de comunicação a longa distância. Essa convergência fica clara com a TV DIGITAL que veio para revolucionar e informar o dia-a-dia dos cidadãos.

Cada vez mais os processos industriais passam por aperfeiçoamentos com objetivo de aumentar a produção, reduzir custos, reduzir acidentes, melhorar a qualidade do produto final. É com essa necessidade que a automação industrial, auxiliada pela instrumentação, ganha seu espaço na indústria e passa a ser um diferencial entre o pequeno e o grande, entre o bom produto e o produto excelente.

A integração de sistemas de comunicação em redes (*intranet/internet*) tem favorecido consideravelmente o aumento da produção das empresas atendendo a todos os pré-requisitos de produção, qualidade e segurança. Hoje as grandes empresas possuem redes corporativas onde são feitos treinamentos, divulgação de informações internas. Tudo isso com objetivo de agilizar a informação.

Outra relação bastante forte é entre a eletrônica e sistemas de comunicação. Isso fica bastante claro ao longo do trabalho, uma vez que são utilizados conhecimentos em controladores lógicos programáveis, redes de computadores, linguagem de programação tudo isso com um único objetivo, automatizar o processo.

Essa parceria tem proporcionado uma evolução tecnológica cada vez mais rápida e eficiente.

O trabalho desenvolvido nessa monografia tem como objetivo principal fazer a supervisão de uma planta de automação industrial através de recursos *Web* (páginas *HTML* e linguagem Java), disponibilizando essa supervisão num acesso via internet ou intranet em qualquer parte da empresa. Esse tipo de monitoramento é rápido, prático e bastante eficaz já que é necessário apenas um computador para que o usuário tenha acesso às informações da planta. Lembrando que para disponibilizar esse tipo de acesso têm que ser feitos mecanismos de controle e segurança para não colocar o processo industrial em risco.

1.1 MOTIVAÇÃO

O trabalho foi proposto pela necessidade de integrar, de forma segura e sem erro, um sistema de automação industrial utilizando CLP (controlador lógico programável) com a intranet/internet. Com isso passaremos a monitorar, através de páginas *HTML*, as variáveis envolvidas nesse sistema.

Vantagens da comunicação via intranet/internet

- Oferecer acesso ao sistema de qualquer parte do mundo;
- Implementar várias tecnologias para acesso a rede (cabo coaxial, par trançado, fibra ótica, rádio, frame relay, *ATM*, *WI-FI*, *WIMAX*, etc...).

Embora a interface seja praticamente determinada pelo CLP utilizado, algumas considerações podem ser feitas para a comunicação via *internet*

- Suporte a comunicação segura, com criptografia
- Suporte a banco dados, linguagens de desenvolvimento de páginas, tais como: *java*;
- Recursos de tecnologia *flash*;
- Facilidade de implementação e manutenção dessas tecnologias;
- Suporte técnico

Há necessidade de estudar as tecnologias envolvidas para esse tipo de integração: fabricantes, modelos de CLPs, recursos oferecidos, sistemas de segurança para acesso, protocolos de comunicação, programas para desenvolver páginas *HTML*.

Na figura Figura 1.1 temos a foto de um modelo de planta industrial.

Figura 1.1 – Modelo de planta industrial - vista lateral

1.2 SITUAÇÃO ANTERIOR

A planta existente (figura 1.1) é projetada para avaliação de tecnologias para medição da vazão, pressão e temperatura.

Ela consiste de um tanque com separador, bomba d'água com velocidade controlada via inverso WEG CFW 09, CLP ALTUS PO3042, módulos de entrada/saída e interface *IHM CIMREX* 12, tubo de venturi. O inversor recebe do CLP o comando para alteração da velocidade. O comando do CLP para ajuste de velocidade é através de linguagem l*adder*.

1.3 METODOLOGIA

O trabalho foi desenvolvido na sala de Automação Industrial do Laboratório de Dispositivos e Nanoestruturas (LDN). Foi disponibilizado um modelo de cada equipamento existente na planta e com isso fizemos a montagem na bancada de testes.

O objetivo foi realizar uma aplicação para supervisão da planta industrial utilizando acesso via *HTTP* através do *WebGate*. Para isso todas as simulações possíveis foram realizadas na bancada de testes.

O desenvolvimento do trabalho foi dividido nas seguintes metas:

META 1 – REVISÃO BIBLIOGRÁFICA

Estudo através de leitura dos manuais, pesquisa na internet sobre as ferramentas existentes e como integra-las com o CLP. Nesta etapa foi feito estudo teórico das ferramentas que serão utilizadas no projeto para entender seu funcionamento e as formas de interligá-las fazendo com que trabalhem integradas. Ferramentas que foram estudadas: *WEBGATE*, *WEBVIEW*, CLP PO3042, *IHM CIMREX* 12, INVERSOR CFW 09.

META 2 – ATIVAÇÃO DO *WEBGATE* PARA TESTES

Fazer a configuração do módulo *WEBGATE* para que ele possa ser acessado através da rede

Transferir as bibliotecas Java

Testar o funcionamento básico

META 3 – TESTE DE FUNCIONAMENTO

Implementar aplicações industriais utilizando os *JAVA*, *CLP* e *FireWall* na bancada de teste.

Avaliar possíveis falhas e não conformidades

Testar desempenho de funcionamento para posterior validação

Testar CPU PO3045 que possui duas entradas seriais RS232.

META 4 – AVALIAR RESULTADOS

Avaliar resultados do desempenho

Fazer ativação na planta industrial

1.4 SEGURANÇA

A comunicação segura consiste em proteger a informação e os recursos da rede. Ela também trata na detecção de falhas na comunicação, ataques à infra-estrutura e reação a esses ataques.

Um sistema conectado a uma rede está sujeito a diversas formas de invasão e tentativas de ataque. Para combater isso são necessários sistemas de segurança.

Propriedades desejáveis em uma comunicação segura:

a) Confidencialidade: Somente remetente e destinatário devem entender o conteúdo da mensagem transmitida.
b) Autenticação: O remetente e o destinatário precisam confirmar a identidade da outra parte envolvida na comunicação.
c) Integridade e não repudiação de mensagem: Mesmo que o remetente e o destinatário consigam se autenticar reciprocamente, eles também querem assegurar que o conteúdo de sua comunicação, não seja alterado por acidente ou por má intenção durante a transmissão.

Criptografia

A técnica da criptografia permite que um remetente disfarce os dados de modo que um intruso não consiga obter nenhuma informação dos dados disfarçados. O destinatário deve estar habilitado a recuperar os dados originais a partir dos dados criptografados.

Autenticação

Existem diversas formas de se obter segurança nos equipamentos. Uma forma seria o próprio sistema embutido no equipamento, que possui cadastro de usuários, controle de acesso, hierarquias de nível. Porém, isso não é suficiente para deixar o acesso ao sistema mais seguro.

Uma implementação mais segura seria através de *firewall* que é uma combinação de hardware e software que isola a rede interna de uma organização da internet em geral, permitindo a passagem de alguns pacotes e bloqueio de outros.

Há dois tipos de *firewalls*: *firewalls* de filtragem de pacotes (que funcionam na camada de rede) e *gateways* de camada de aplicação (que funcionam na camada de aplicação).

A Figura 1.2 mostra uma forma de ligação para o firewall. Na Figura 1.3 mostramos um exemplo para sistema de comunicação utilizando roteadores, *links*, *firewall*, *WebGate* e CLPs.

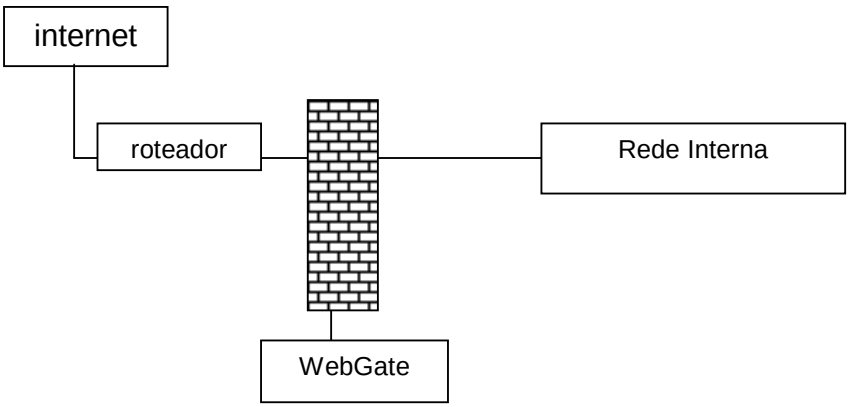


Figura 1.2 - ligação do *WebGate* com a internet através de *firewall*

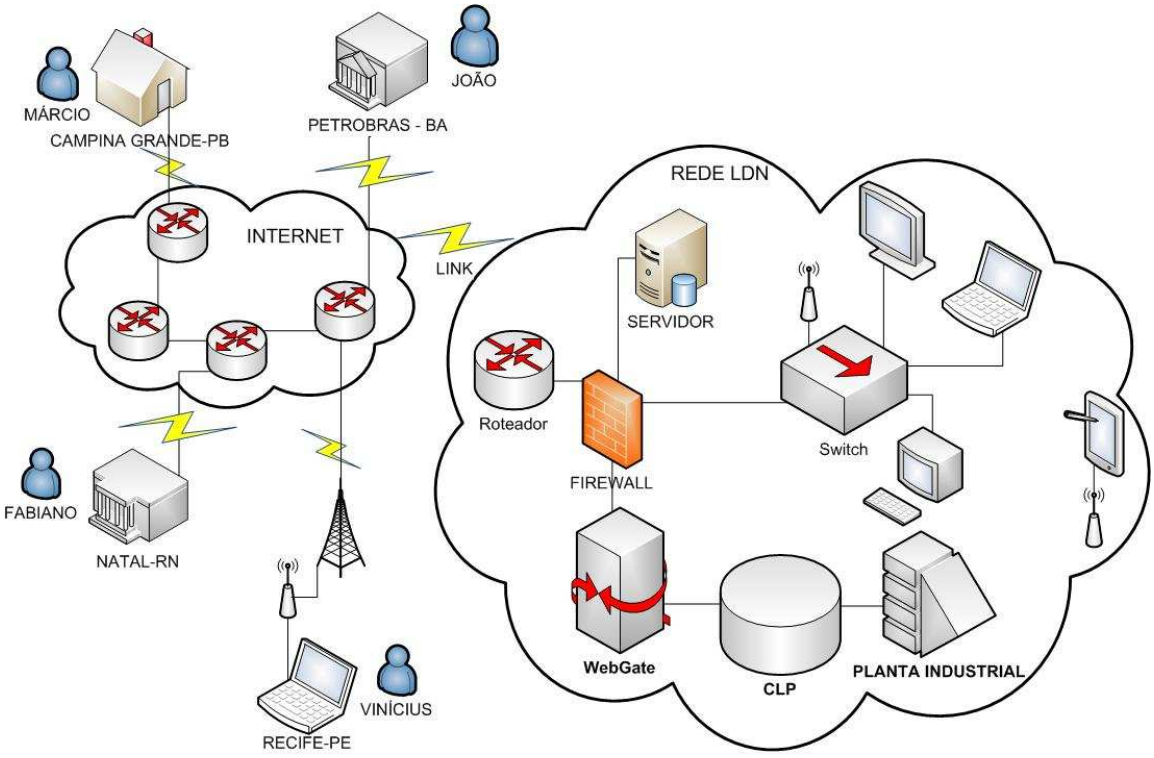


Figura 1.3 – Sistema de comunicação com *firewall e WebGate*

Neste capítulo vimos os motivos para os quais estamos desenvolvendo esse trabalho, a metodologia utilizada para que as atividades sejam executadas no prazo e os aspectos necessários mínimos de segurança exigido na elaboração do projeto. No capítulo 2 veremos as tecnologias que iremos utilizar para realizar nossas metas. .

CAPÍTULO 2

TECNOLOGIAS ENVOLVIDAS

A implementação de um sistema de automação e controle permitiu o estudo e utilização de diversas tecnologias, entre elas pode-se ter:

a) CLP ALTUS PO3042 e PO3045 - UCP
b) MÓDULO ALTUS PO2022 16DOR
c) MÓDULO ALTUS PO1112 8AI
d) MODULO ALTUS PO2132 4AO
e) MODULO WEG CFW 09
f) Interface IHM CIMREX 12
g) Sensor de pressão piezoresistivo – PCB *Piezotronics*
h) Motor trifásico WEG
i) *WEBGATE* PO9900
j) WEBVIEW
k) REDE ETHERNET 10/100
l) Protocolos *TCP/IP*, ALNET (I e II), http, *FTP*
m) Linguagem de programação (*HTML*, *Java Script*, *Java*, *XSL*, *XML*)

2.1 CONTROLADOR LÓGICO PROGRAMÁVEL

Atualmente o Controlador Lógico Programável, também chamado de CLP, ou ainda de PLC (*Programmable Logic Controller*), ou simplesmente CP (Controlador Programável) é uma ferramenta bastante utilizada na indústria para o controle de máquinas e/ou processos devido a sua flexibilidade e estabilidade.

2.1.1 Histórico

O Controlador Lógico Programável nasceu dentro da General Motors, em 1968, devido a grande dificuldade de mudar a lógica de controle dos painéis de comando a cada mudança na linha de montagem. Tais mudanças implicavam em altos gastos de tempo e dinheiro.

Sob a liderança do engenheiro Richard Morley, foi preparada uma especificação que refletia as necessidades de muitos usuários de circuitos e relés, não só da indústria automobilística como de toda a indústria manufatureira [14].

Nascia assim um equipamento bastante versátil e de fácil utilização, que vem se aprimorando constantemente, diversificando cada vez mais os setores industriais e suas aplicações.

2.1.2 Principio de funcionamento

O Processador lê dados de entrada de vários dispositivos, executa o programa do usuário armazenado na memória e envia dados de saída para comandar os dispositivos de controle. Este processo de leitura das entradas, execução do programa e controle das saída é feito de uma forma contínua sendo chamado de ciclo de varredura [11].

A UCP (Unit Central Processing ou Unidade Central de Processamento) coordena as atividades do sistema, interpretando e executando um conjunto de instruções, conhecido como programa Executivo ou Monitor que é um programa exclusivo para monitoramento e controle do CP.

O Controlador Programável opera executando uma seqüência de atividades definidas e controladas pelo programa Executivo. Este modo de operação ocorre em um ciclo, chamado de Ciclo de Varredura ("Scan"), que consiste em:

- Leitura das entradas externas;
- Execução da lógica programada;
- Atualização das saídas externas.

Na fase de leitura das entradas, o Processador endereça o sistema de E/S, obtém os estados dos dispositivos que estão conectados, e armazena estas informações na forma de bits "1" ou "0", dependendo do estado obtido (ponto energizado equivale ao binário "1" e ponto desenergizado ao binário "0"). A região da memória utilizada para armazenar estas informações é chamada de Tabela Imagem das Entradas - TIE.

Na fase de execução da lógica programada pelo usuário, a UCP consulta a TIE para obter os estados dos dispositivos. Nesta fase, os resultados das lógicas programadas cujas saídas tenham um ponto correspondente no rack de saída são armazenados em uma área de memória que é chamada de Tabela Imagem das Saídas - TIS. As lógicas que possuem saídas internas serão armazenadas nas áreas correspondentes. Durante a execução da lógica programada, se for necessário a referência a uma saída qualquer, dentro do mesmo ciclo, esta tabela é consultada.

2.1.3 Componentes básicos

Todos os controladores programáveis, independentes de marca, tamanho, complexidade, possuem os mesmos componentes básicos: Processador, memória, sistemas de entrada/saídas, fonte de alimentação e terminal de programação.

As três partes principais (processador, memória e fonte de alimentação) formam o que chamamos de UCP.

2.1.4 Vantagens

Os CLPs oferecem um considerável número de benefícios para aplicações na indústria. Estes benefícios podem resultar em economia que excede o custo do CLP em si e devem ser considerados quando da seleção de um dispositivo de controle industrial. As vantagens da utilização de CLPs, se comparados a outros dispositivos de controle industrial incluem:

- Menor ocupação de espaço;
- Baixa potência elétrica requerida;
- Reutilização;
- Programável, se ocorrerem mudanças de requisitos;
- Confiabilidade;
- Manutenção fácil;
- Flexibilidade, satisfazendo um número maior de aplicações;
- Permite a interface através de rede de comunicação com outros CLPs e com microcomputadores;
- Projeto rápido de sistemas.

2.1.5 Desvantagens

Devido a grande variedade de fabricantes e modelos de CLPs, temos algumas desvantagens:

- Tempos de varredura. Alguns CLPs demoram mais que outras em seu processamento;
- Tipos de operandos. Alguns modelos só trabalham com operandos de memória, auxiliares e decimais;
- Limite de portas seriais;
- Ausência de linguagem estruturada (ST);
- Falta de um simulador para testar apenas a lógica de programação sem precisar o equipamento está ligado.

2.1.6 Elementos da arquitetura ponto

A Série Ponto é um sistema de controle distribuído com E/S remotas. Possui uma arquitetura flexível que permite os acessos a módulos remotos via diferentes padrões de redes de campo. Na Tabela 2.1 são apresentados os modelos e na Figura 2.1 os modelos montados no trilho.

Tabela 2.1– Modelos da arquitetura ponto

Modelo	Descrição
PO 1xcc	Módulos de entrada
PO 2xcc	Módulos de saída
PO 3xcc	UCPs - Unidades Centrais de Processamento
PO 4xcc	Interfaces de rede de Campo
PO 5xcc	Cabeças de Redes de Campo
PO 6xxx	Bases
PO 7xcc	Módulos especiais
PO 8xcc	Fontes e acessórios
PO 9xxx	Software para UCPs
PO 6000	Base E/S Digital mola
PO 6001	Base E/S analógica mola
PO 6050	Base E/S Digital parafuso
PO 6051	Base E/S analógica parafuso
PO 6101	Base E/S analógica mola com fusíveis
PO 6150	Base E/S Digital mola com fusíveis
PO 6151	Base E/S analógica mola com fusíveis

Figura 2.1- Elementos da série ponto

Abaixo descreveremos apenas os elementos que foram utilizados no desenvolvimento deste trabalho.

Trilho de Montagem

A Série Ponto é montada em trilhos padrão DIN TS35. A fixação dos módulos nos trilhos se dá por encaixe.

UCP

É responsável pela execução das funções de controle, realizando o ciclo básico de leitura dos pontos de entrada, execução do programa aplicativo, atualização das saídas e comunicação com o sistema supervisório, entre outras funções.

Fonte

Provê alimentação para a UCP e todos os módulos de E/S locais. Pode ser utilizada como fonte de expansão quando for necessário mais corrente nos segmentos do barramento. As UCPs de menor porte possuem a fonte integrada.

Barramento

Um sistema é constituído por um barramento Local, formado por uma UCP e seus módulos de E/S, e uma série de barramentos remotos, composto pela Cabeça de Rede de Campo e módulos de E/S.

O barramento local comporta até 30 módulos de E/S divididos em até 4 segmentos. Os barramentos remotos, por sua vez, possuem outras limitações relacionadas com a quantidade de dados a serem transmitidos no protocolo escolhido.

Bases

As bases são elementos modulares que formam os barramentos. São montadas em trilhos TS35 e distribuem assim a alimentação, sinais do barramento e sinais das E/S para os módulos da série. As bases possuem os bornes - nas opções mola ou parafuso - para ligação com a fiação de campo e, opcionalmente, fusíveis de proteção. A escolha da base está relacionada ao tipo de módulo a ser utilizado.

Módulos de E/S

Os módulos de E/S são encaixados nas bases, e tem a função de adaptar os diferentes tipos de sinais de campo e enviá-los à UCP ou Cabeça de Rede. Apresentam uma grande variedade de tipos e faixas de operação, para cobrir as mais diferentes necessidades. Os módulos podem ser trocados à quente, sem necessidade de desconectar cabos ou desligar todo o sistema.

A alimentação dos circuitos de campo deve ser provida por fontes externas.

Terminação

Tem a função de casar a impedância do barramento de comunicação de um barramento local ou remoto. A terminação é um conector que deve ser instalado na última base do barramento. Este componente é fornecido juntamente com a base da UCP e da cabeça de rede de campo.

Configuração

Uma etapa muito importante é a definição do tipo de configuração a ser utilizada pelo CLP.

Limites

No caso de uma UCP com seu barramento local, os seguintes limites devem ser respeitados:

- Número máximo de módulos por segmento: 10
- Número máximo de segmentos: 4
- Número máximo total de módulos: 30
- Número máximo de módulos alimentados por uma fonte: 12, distribuídos no máximo em dois segmentos. Este limite de 12 módulos pode ser

ultrapassado se indicado pelo software ProPonto, que calcula o limite usando o consumo específico de cada módulo.

Na tabela 2.2 mostramos algumas das opções disponíveis.

Tabela 2.2 – Tipos disponíveis de UCP

	PO5063	**PO3042**	**PO3045**	**PO3145**	**PO3245**	**PO3345**
Denominação	Cabeça de Rede de Campo PROFIBUS	UCP 128K Flash, 30 Módulos E/S, 1 Interface Serial RS232	UCP 128 K Flash,, 16 Módulos E/S, 2 Interfaces Seriais RS232	UCP 256 K Flash, 30 Módulos E/S, 3 Interfaces Seriais 2 - RS232 1 - RS485 MODBUS	UCP 512 K Flash, 16 Módulos E/S, 3 Interfaces Seriais 2 - RS232 1 - RS485 Inteface Ethernet TCP/IP	UCP 512 K Flash, 30 Módulos E/S 3 Interfaces Seriais 2 - RS232 1 - RS485 Interface Ethernet TCP/IP com WebServer
Base Compatível	PO6500	PO6500	PO6305	PO6305	PO6305	PO6305
Base de Fonte	não necessita	não necessita	não necessita	PO6800	PO6800	PO6401 PO6401

Programação

A programação dos CLPs Altus é feita com o sistema MásterTool Programming MT 4100 3.81. A linguagem utilizada na programação é o *LADDER* (linguagem baseada em contatos de relés)

a) Elementos de programação

Um programa aplicativo é composto por 4 elementos básicos:

- módulos
- lógicas
- instruções
- operandos

Um programa aplicativo é composto por diversos módulos, permitindo uma melhor estruturação das rotinas de acordo com as suas funções. Os módulos são programados em linguagem de relés, seguindo a tendência mundial de normalização nesta área.

Um módulo de programa aplicativo é dividido em lógicas de programação. O formato de uma lógica de programa aplicativo utilizado nos CLPs das séries AL-600, AL-2000, AL-3000 e QUARK e PICCOLO permite até oito elementos em série e até quatro caminhos em paralelo.

As instruções são utilizadas para executar determinadas tarefas por meio de leituras e/ou alterações do valor dos operandos.

Os operandos identificam diversos tipos de variáveis e constantes utilizadas na elaboração de um programa aplicativo, podendo ter seus valores modificados de acordo com a programação realizada. Como exemplo de variáveis pode-se citar pontos de E/S e memórias contadoras.

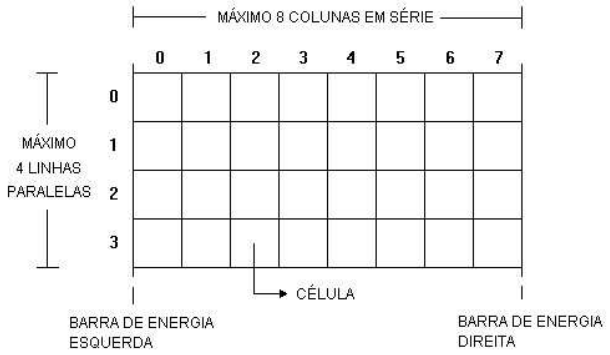
Organização de memória dos CLPs Altus

O programa aplicativo é armazenado no controlador em uma área de memória dividida em bancos. Podem existir um ou mais bancos de memória RAM e EPROM, conforme o modelo do CLP e a sua configuração de memória, cada banco possuindo 16, 32 ou 64 Kbytes. A memória EPROM pode existir na forma de cartucho removível ou como flash EPROM interna ao CLP.

Lógica

Chama-se lógica a matriz de programação (figura 2.2) formada por 32 células (elementos da matriz) dispostas em 4 linhas (0 a 3) e 8 colunas (0 a 7). Em cada uma das células podem ser colocadas instruções, podendo-se programar até 32 instruções em uma mesma lógica.

Cada lógica presente no programa simula um pequeno trecho de um diagrama de relés real.



	0	1	2	3	4	5	6	7
0	1	5	9	13	17	21	25	29
1	2	6	10	14	18	22	26	30
2	3	7	11	15	19	23	27	31
3	4	8	12	16	20	24	28	32

Figura 2.2 - Matriz de programação

As duas linhas laterais da lógica representam barras de energia entre as quais são colocadas as instruções a serem executadas.

OPERANDOS

São elementos utilizados para elaboração de programas. Eles podem definir valores constantes ou variáveis e são identificados por endereços

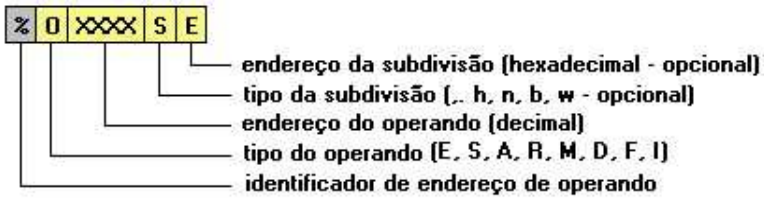
A identificação e utilização de um operando pelo seu endereço são caracterizadas pelo uso do caractere % como primeiro caractere do nome.

Exemplos:
%A0000.0, %TD0001, %E0000.2, %M0004, %M0000*A

OPERANDOS SIMPLES

Os operandos simples são utilizados como variáveis de armazenamento de valores no programa aplicativo. Conforme a instrução que os utilizam, eles podem ser referenciados na sua totalidade ou em uma subdivisão (uma parte do operando). As subdivisões de operandos podem ser palavra, octeto, nibble ou ponto.

Formato geral de um operando simples:


% O XXXX S E
endereço da subdivisão (hexadecimal - opcional)
tipo da subdivisão (. h, n, b, w - opcional)
endereço do operando (decimal)
tipo do operando (E, S, A, R, M, D, F, I)
identificador de endereço de operando

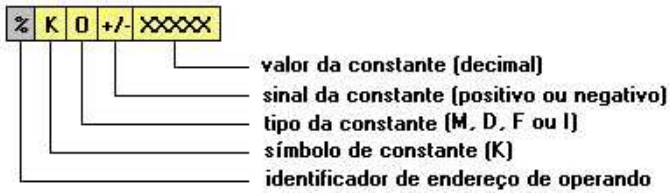
Tipo do operando:

%E entrada	%S saída	%A auxiliar	%R endereço no barramento	%M memória	%D decimal	%F -real	%I - inteiro

OPERANDOS CONSTANTE

Os operandos constante são utilizados para a definição de valores fixos durante a edição do programa aplicativo.

Formato geral de um operando constante:


% K O +/- XXXX
valor da constante (decimal)
sinal da constante (positivo ou negativo)
tipo da constante (M, D, F ou I)
símbolo de constante (K)
identificador de endereço de operando

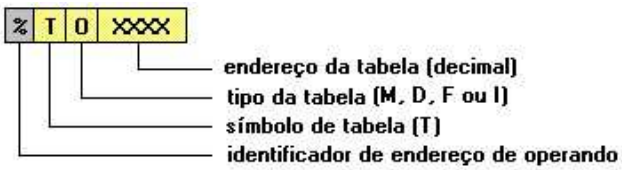
Tipo da constante:

%M memória	%D decimal	%F real	%I inteiro

OPERANDOS TABELA

Tabelas de operandos são conjuntos de operandos simples, constituindo arranjos unidimensionais. São utilizados índices para determinar a posição da tabela que se deseja ler ou alterar.

Formato geral de um operando tabela:



Tipo da tabela:

%TM memória	%TD decimal	%TF real	%TI inteiro

b) Tipos de módulos

É extremamente importante definir os tipos de módulos que irão compor o programa. Caso isso não seja feito, poderá haver diversos problemas durante a execução do CLP. A Figura 2.3 mostra a tela de criação dos módulos.

Existem quatro tipos de módulos de programa.

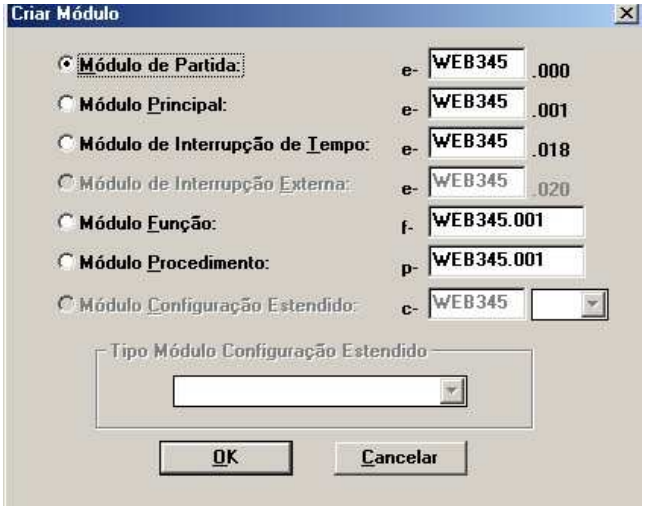


Figura 2.3 - Tela de módulos de criação do projeto

Módulo C (Configuração): existe um módulo de configuração por programa aplicativo, contendo os parâmetros de configuração do CLP (C000).

Módulo E (Execução): podem existir até 4 módulos de execução por programa aplicativo. Os mesmos são chamados somente pelo sistema operacional do CLP (E000, E001, E018 e E020).

- **E000** - Módulo de inicialização
- É executado uma única vez, ao se energizar o CLP ou na passagem de modo programação para execução com o MASTERTOOL, antes da execução cíclica do módulo E001.
- **E001** - Módulo seqüencial de programa aplicativo
- Contém o trecho principal do programa aplicativo, sendo executado ciclicamente.
- **E020** - Módulo acionado pela entrada de Interrupção
- **E018** - Módulo Acionado por Interrupção de Tempo

O trecho de programa aplicativo colocado neste módulo é chamado para a execução em intervalos de tempo periódicos. Após o seu final, o sistema retorna a execução para o ponto do processamento seqüencial onde o módulo havia sido interrompido.

Módulo P (Procedimento): podem existir até 116 módulos procedimento por programa aplicativo. Eles contêm trechos de programa aplicativo, sendo chamados por instruções colocadas

em módulos de execução, procedimento ou função. Após serem executados, o processamento retorna para a instrução seguinte à de chamada. Os módulos P funcionam como sub-rotinas, não permitindo a passagem de parâmetros para o módulo chamado (P000 a P115).

Módulo F (Função): podem existir até 116 módulos função por programa aplicativo. Eles contêm trechos de programa aplicativo escritos de forma genérica, permitindo a passagem de parâmetros para o módulo chamado, de forma a poderem ser reaproveitados em vários programas aplicativos diferentes. São semelhantes as instruções, podendo ser chamados por módulos de execução, procedimento ou função. (F000 a F115).

c) Configuração via proponto

Outra etapa importante é a configuração dos módulos do CLP através do sistema Proponto. É nele que são configurados: tipo de UCP, fonte, módulos de E/S. bases, de acordo com a montagem prática.

d) Estados de operação

Existem cinco estados ou modos de operação do CLP: inicialização, execução, programação, ciclado e erro. O estado em que o controlador programável se encontra é indicado nos LEDs do painel frontal da UCP, podendo também ser consultado pelo MasterTool, através da caixa de diálogo Estado.

- Estado Inicialização
- Estado Execução
- Estado Programação
- Estado Ciclado
- Estado de Erro

Em condições normais, o controlador programável pode estar nos modos execução, programação e ciclado, sendo esses modos acionados através de comandos do MasterTool. Na ocorrência de alguma situação de funcionamento errôneo nestes modos, o CLP passa para estado de erro.

A recuperação do modo erro somente é possível passando-se o controlador programável para modo programação. A Figura 2.4 apresenta as possibilidades de troca de estados.

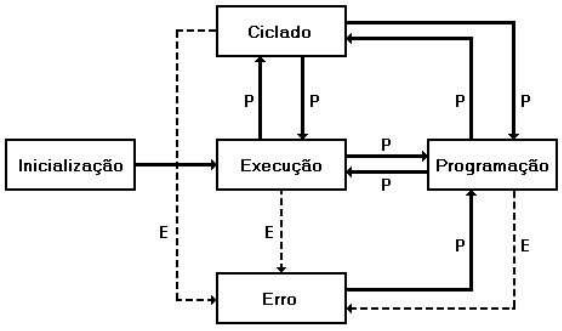


Figura 2.4 – Modos de operação do CLP

P – comando do máster Tool
E – Situação de erro

Nos modos execução e programação é possível carregar e ler módulos R pelo canal serial do roteador, bem como monitorar e forçar quaisquer operandos utilizados.

Essas operações não são possíveis caso o CLP esteja em modo erro.

e) Execução do projeto

Quando o CLP é energizado ou após a passagem para modo execução, as inicializações do sistema são realizadas de acordo com o conteúdo do módulo C, sendo logo após executado o módulo E000 uma única vez. O controlador programável passa então para o processamento cíclico do módulo E001, atualizando as entradas e saídas e chamando o módulo E018, quando existente, a cada período de tempo de interrupção programado. A Figura 2.5 mostra esquematicamente a execução do programa aplicativo.

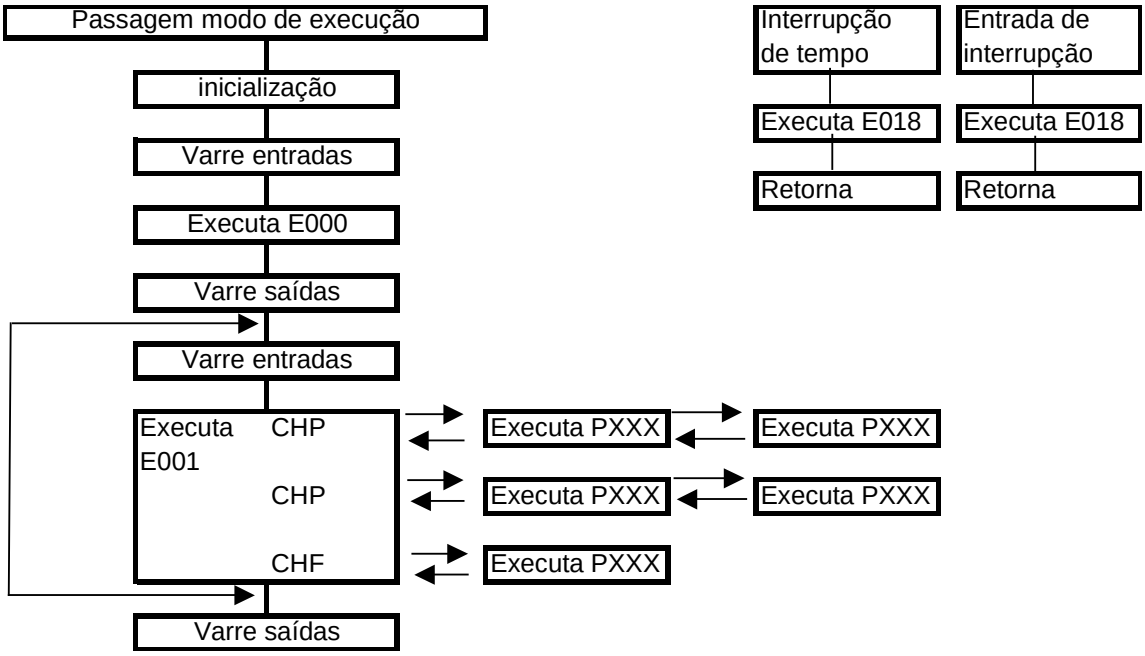


Figura 2.5 – Processo de execução de um programa

f) Led´s de diagnóstico

Podemos ver na Tabela 2.3 os leds de diagnóstico do CLP. Eles têm objetivos de orientar o operador nos estados em que o CLP está trabalhando. Caso o CLP saia de seu funcionamento normal, um dos leds irá diagnosticar. Na Tabela 2.4 temos os modos de execução.

Tabela 2.3 – LED´S de diagnóstico da UCP PO3045

LED Estado Significado	Estado	Significado
EX	Execução	Indica que a UCP está executando o programa aplicativo corretamente. Normalmente, neste estado, o equipamento se encontra varrendo as entradas e atualizando as saídas de acordo com a lógica programada.
PG	Programação	Indica que a UCP está em modo de programação. Neste estado, o equipamento fica aguardando os comandos enviados pelo programador, sem executar o programa Aplicativo
DG	Diagnóstico	Em combinação com outros LEDs, indica várias condições diferentes. Para obter mais informações, consulte o item Diagnósticos via Painel na seção Manutenção.
ER	Erro	Erro Este LED indica que o processador da UCP detectou alguma anormalidade no funcionamento do seu hardware ou software.
WD	Watchdog	Indica que o circuito de cão-de-quarda está acionado. Este circuito monitora continuamente a execução do microcontrolador principal da UCP, desabilitando-o em caso de falhas.
BT	Bateria	indica que a bateria que mantém a memória não-volátil (operandos retentivos) está com a tensão abaixo do normal e deve ser substituída.
COM1 TX	Transmite	Indica que a UCP está transmitindo bytes no canal serial da rede ALNET I (conector COM1 da base).
COM1 RX	Recebe	Indica que a UCP está recebendo bytes no canal serial da rede ALNET I (conector COM1 da base).
COM2 TX	Transmite	Indica que a UCP está transmitindo bytes no canal serial RS-485 (conector COM2 da base). Disponível somente na UCP PO3145.
COM2 RX	Recebe	Indica que a UCP está recebendo bytes no canal serial RS-485 (conector COM2 da base). Disponível somente na UCP PO3145.
COM3 TX	Transmite	Indica que a UCP está transmitindo mensagem no terceiro canal serial RS-232 (conector COM3 da base).
COM3 RX	Recebe	Indica que a UCP está recebendo mensagem no terceiro canal serial RS-232 (conector COM3 da base).

Tabela 2.4 – Modos de execução da UCP

EX	PG	DG	ER	WD	MODO DE OPERAÇÃO
LIGADO	LIGADO	LIGADO	LIGADO	DESLIGADO	Inicialização
LIGADO	DESLIGADO	-	DESLIGADO	DESLIGADO	Execução
LIGADO	LIGADO	-	DESLIGADO	DESLIGADO	Ciclado
DESLIGADO	LIGADO	-	DESLIGADO	DESLIGADO	Programação
DESLIGADO	DESLIGADO	-	LIGADO	DESLIGADO	Erro

g) MÓDULO PO1112

Módulo com 8 entradas analógicas para medição de tensão, corrente, resistência, RTD e termopares. No apêndice A encontra-se um diagrama de ligações elétricas do módulo.

h) MÓDULO PO2132

Módulo com quatro saídas analógicas para tensão e corrente.

2.2 Inversor de frequência CFW09

CFW09 (Figura 2.6) é um inversor de freqüência cujo objetivo e converter tensão alternada (ca) em tensão contínua(cc). Ele possui:

- 6 entradas digitais programáveis;
- 2 entradas analógicas programáveis;
- 2 saídas analógicas programáveis;
- 1 IHM para configuração local e declaração de parâmetros;

Figura 2.6 - Módulo Inversor WEG CFW09

2.3 Interface IHM Cimrex 12

A interface CIMREX 12 (Figura 2.7) é utilizada para realizar a supervisão e controle local. Os programas para monitoramento são desenvolvidos e enviados do computador para a IHM através de cabo serial.

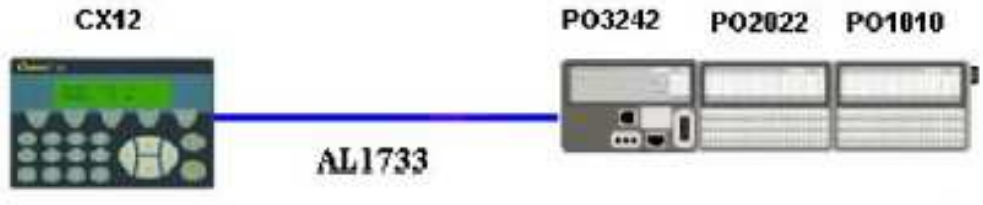


Figura 2.7 – IHM CIMREX 12

2.4 ETHERNET

A arquitetura ethernet é a mais utilizada em redes locais, sendo o padrão que define como os dados serão transmitidos fisicamente através dos cabos de rede. O papel do ethernet é pegar os dados entregues pelo protocolo TCP/IP e inseri-los dentro de quadros que serão enviados através da rede.

2.5 *WEBGATE*

A interface de comunicação *WebGate*, PO9900, permite que controladores programáveis Altus e outros equipamentos com protocolo ALNET I escravo possam ser conectados a uma rede Ethernet.

O *WebGate* permite que esses controladores comuniquem-se com outros CLPs, softwares de supervisão, com o programador MasterTool e com um browser.

2.5.1 Características da interface PO9900

- Interface Ethernet 10baseT para conexão local e internet
- Servidor de páginas (*WebServer*) no formato *HTML*
- *Firewall*
- Controle de acesso de usuários via *FTP*
- Integração de qualquer CLP Altus à rede *Ethernet*, através de ALNET II/IP ou *HTML/XML*;
- O *WEBGATE* permite o acesso ao equipamento ao qual está ligado das seguintes formas:
- Protocolo ALNET II/IP
- Browser utilizando protocolo *HTTP*

2.5.2 Alimentação

O *WebGate* é alimentado por uma fonte (AL 1532) cuja a saída é 24Vdc e que tem como entrada 220Vac. Na Figura 2.8 é exibido o diagrama de alimentação do WebGate.

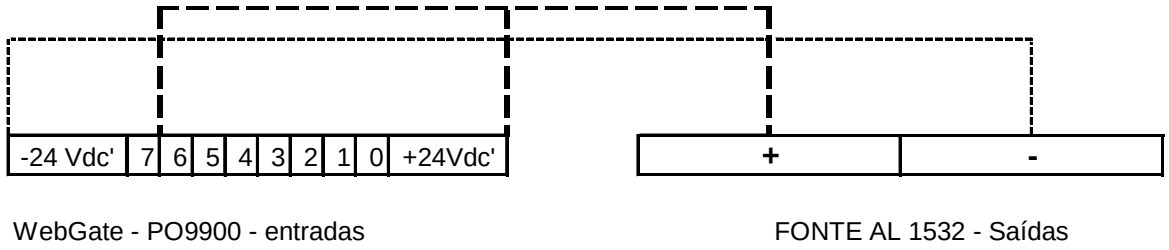


Figura 2.8 – Diagrama de alimentação do *WebGate* PO9900

2.5.3 Servidor web

O *WebGate* vem com um servidor *Web* para armazenamentos de páginas *HTML*, *applets Java*. Ele possui páginas prontas dentro do diretório *WWW*, mas o usuário pode criar suas próprias páginas e adicioná-las dentro do diretório através de programa *FTP*.

2.5.4 Acesso ao W*ebGate*

Para acessar o *WebGate* via rede intranet/internet é necessário fazer as configurações de rede informando: endereço *IP*, máscara de rede, *DHCP*. A configuração inicial é feita através de porta serial utilizando o programa Hyper Terminal.

2.5.5 Conexão do W*ebGate* com o CLP

A conexão entre o *WebGate* e o CLP é feita via rede ALNET I, utilizando-se os cabos PO8500 e AL-1726. Esses cabos são para UCP PO3042 e *WebGate* PO9900. Para se realizar a conexão, é necessário o conversor TTL serial do cabo PO8540, onde uma extremidade contém um conector RJ11 e a outra extremidade é um conector serial macho. O conector RJ11 deve ser conectado à porta COM2 do *WebGate*, enquanto que o DB9 deve ser ligado a porta ALNET I.

Outra forma de conexão é utilizando os cabos AL 1330 e AL 1390, conforme Tabela 2.5.

Tabela 2.5 – cabos de conexão

CABO	Descrição	UCP – SÉRIE
AL 1330	Este cabo possui conector DB9 RS232 macho e outro DB9 fêmea padrão IBM PC	PÍCOLO
AL 1390	Este cabo possui conector DB9 macho com pinagem padrão e outro DB9 fêmea padrão IBM PC	AL 2000, QK 600, QK 801

2.5.6 Conexão com a rede ethernet

A conexão com a rede Ethernet é feita através de cabo par trançado (10baseT) deve ser conectado um cabo ethernet à porta NET do *WebGate*, ligando-se a outra extremidade ao HUB da rede ethernet. Outra forma de conexão é através de cabo cruzado *(crossover*) interligado o equipamento diretamente ao computador**.**

2.6 PROTOCOLOS

São regras e instruções utilizadas na comunicação entre HARDWARE e SOFTWARE. Na Tabela 2.6 podemos ver os principais protocolos utilizados na comunicação com a rede *intranet/internet* e na Tabela 2.7 comparação entre os modelos.

Tabela 2.6 – Tipos de protocolo

ALNET I	protocolo ALTUS para comunicação wegate - CLP
ALNET II/IP	protocolo ALTUS para comunicação wegate - PC
HTTP	Protocolo para acesso WWW
FTP	protocolo de transferência de arquivos
TCP	protocolo para controle de transferência
IP	protocolo de internet

Tabela 2.7 Comparação entre os modelos OSI, TCP/IP e ALTUS

Camadas	Modelo OSI/ISO	Modelo TCP/IP	Protoclos	Equipamento	ALTUS
7	Aplicação	Aplicação	HTTP FTP	PO3045	Alnet I - RS232 Alnet II
6	Apresentaçào				
5	Sessào			WebGate	HTTP FTP TELNET
4	Transporte	Transporte	TCP	WebGate	TCP
3	Rede	Internet	IP		IP
2	Enlace	Interface com a rede			ETHERNET
1	Física				

2.6.1 ALNET

a) Transação

Uma transação de comunicação normalmente é composta por uma seqüência de duas mensagens: comando (ou requisição) e resposta.

Estas mensagens são trocadas, normalmente, entre dois nós. O objetivo de uma transação é a execução de um serviço de rede. Existe um nó que requisita o serviço (mestre), e outro que fornece o serviço (escravo ou servidor). Inicialmente, o mestre envia uma mensagem (comando) para o escravo. A seguir o escravo envia uma mensagem para o mestre (resposta).

No comando, o mestre pode informar para o escravo:

- O tipo de serviço que deve ser executado
- Pacotes de dados que devem ser recebidos pelo escravo (exemplo: o mestre deseja escrever variáveis no escravo)

Na resposta, o escravo pode informar para o mestre:

- Que o serviço solicitado foi reconhecido e executado
- Que o serviço solicitado não foi reconhecido, ou que não foi executado por algum motivo.
- Pacote de dados requisitado pelo mestre (exemplo: o mestre deseja ler variáveis do escravo)

Uma transação é finalizada quando o nó requisitor recebe: (a) pacote de confirmação de serviço, para o caso de serviços sem resposta; (b) pacote de dados para serviços com resposta de um pacote; (c) o último pacote de dados de serviços com resposta em mais de um pacote; e (d) por *time-out.*

b) Protocolo de comunicação

Uma arquitetura envolvendo comunicação Ethernet TCP/IP pode possuir diferentes protocolos no seu nível de aplicação. Assim sendo, fala-se em Ethernet para o nível de rede, mas em protocolo ALNET II ou MODBUS TCP para o nível de aplicação.

A maioria das interfaces de comunicação baseadas em Ethernet implementam apenas cinco das sete camadas definidas pelo modelo OSI/ISO:

• 10/100Base-TX: camada física;
• Ethernet: camada de enlace;
• Internet Protocol (IP): camada de rede;
• Transport Control Protocol (TCP): camada de transporte;
• ALNET II ou MODBUS TCP: camada de aplicação.

c) Limites de dados por transação ALNET II

Comunicações com os CLPs Altus podem ser realizadas através de comandos de leitura ou escrita de blocos de dados limitados em 220 bytes de dados. Determinados drivers de comunicação podem executar automaticamente o agrupamento de operandos para aproveitar todo o limite do bloco, outros não, devendo ficar a cargo do usuário o agrupamento dos dados de forma manual.

No entanto, para qualquer driver, os limites para estas transações podem variar caso se esteja utilizando os dispositivos *WebGate* ou *WebGate* Plus – PO9900 ou PO9901 – uma vez que para este, os blocos de dados devem ser definidos com até 128 bytes.

O detalhamento do número máximo de operandos que podem ser lidos ou escritos em cada bloco de dados, utilizando-se as interfaces Ethernet Altus é apresentado na Tabela 2.8.

Tabela 2.8 – Número máximo de operandos por bloco de dados

Tipo de Operando	**N° de Bytes por Operando**	**N° Máximo de operandos por bloco para AL-3405, AL-3412, AL-3414, PO7091 e PO7092**	**N° Máximo de operandos por bloco para PO9900 e PO9901**
Octetos de E/S	1 byte	220 operandos	64 operandos
Auxiliar	1 byte	220 operandos	64 operandos
Memória	2 bytes	110 operandos	64 operandos
Decimal, Real e Inteiro	4 bytes	55 operandos	32 operandos
Tabela de Memória	2 bytes por posição	110 posições	64 posições
Tabela de Decimal, Tabela de Real e Tabela de Inteiro	4 bytes por posição	55 posições	32 posições

d) Acesso via protocolo ALNET II/IP

Uma forma de utilização do *WebGate* é como um g*ateway* escravo na rede ethernet, utilizando protocolo ALNET II, e mestre na rede ponto-a-ponto ALNET I.

Nesta configuração o *WebGate* recebe uma requisição ALNET II sobre *TCP/IP* oriunda de um equipamento mestre. Esta requisição é traduzida para o protocolo ALNET I pelo *WebGate* e enviada para o equipamento ligado via interface RS-232C, por exemplo, um CLP. O equipamento, então, responde à requisição ALNET I para o *WebGate* que traduz esta resposta para o protocolo ALNET II sobre o *TCP/IP*, enviando para o mestre que fez a requisição.

Acesso via *WAN*

Na Figura 2.9 qualquer um dos computadores pode ter acesso ao *WebGate*. Um dos computadores esta acessando via intranet local e o outro via rede intranet via rede longa distância que até pode ser a internet, mas que para isso o acesso é possível desde que seja liberado um endereço IP válido.

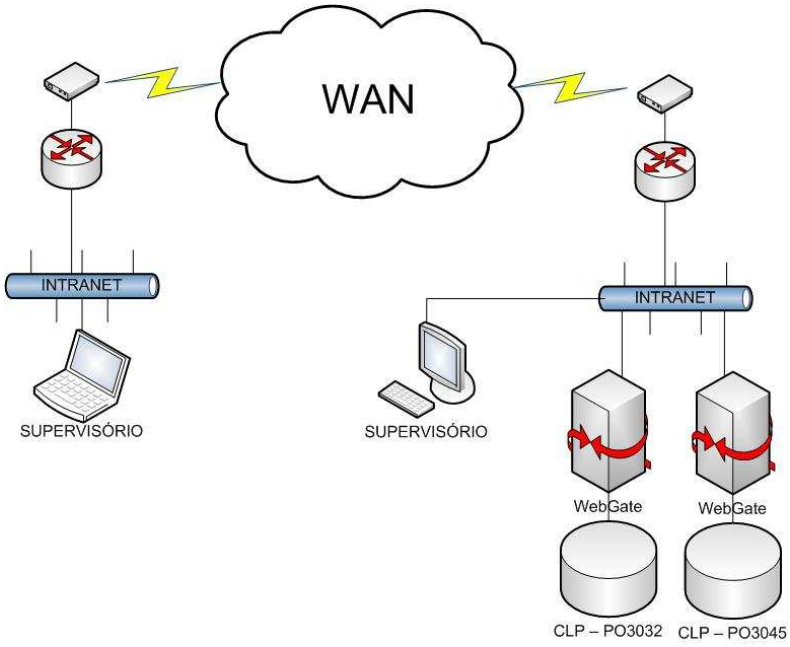


Figura 2.9 – Exemplo de acesso via rede externa e interna
Fonte: Altus Sistemas de Informática S.A. Manual de Utilização do *WebGate* – PO9900. MU209690, 06/2003.

f) Acesso via protocolo ALNET I

É uma rede de comunicação mestre-escravo com transmissão serial para interligação com controladores programáveis (Figura 2.10).

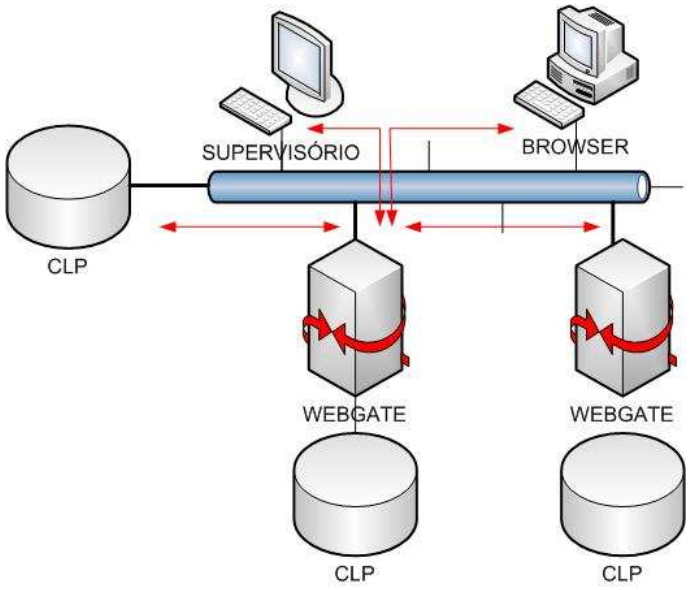


Figura 2.10 – Exemplo de acesso ao CLP via protocolo ALNET I
Fonte: Altus Sistemas de Informática S.A. Manual de Utilização do *WebGate* – PO9900. MU209690, 06/2003.

g) Interface de software

A implementação da pilha de protocolos TCP/IP permite fornecer acesso via Ethernet aos controladores programáveis Altus.

Um módulo de conversão ALNET II <=> ALNET I permite que o equipamento seja utilizado com qualquer controlador programável Altus, além de outros equipamentos que possuam interface com protocolo ALNET I escravo. Na Figura 2.11 temos a interface de acesso ao *WebGate* com todos os protocolos.

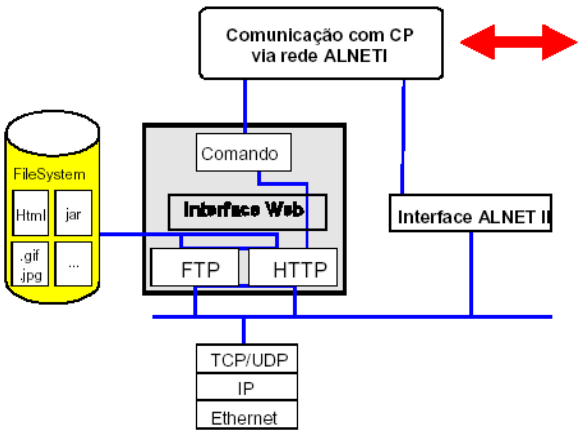


Figura 2.11 – Interface de acesso ao *WebGate*
Fonte: Altus Sistemas de Informática S.A. Manual de Utilização do *WebGate* – PO9900. MU209690, 06/2003.

2.6.2 TCP/IP

É o protocolo mais utilizado nas redes de computadores. Isso se deve a popularização da internet e ao fato do *TCP/IP* ser roteável e possuir arquitetura aberta. Isto é, ele foi criado pensando em redes de longas distâncias onde pode haver vários caminhos para alcançar o destino. A arquitetura aberta é porque cada fabricante pode adotar sua própria versão do *TCP/IP*.

O *TCP/IP* é um conjunto de protocolos dividido em quatro camadas, como pode ser visto na tabela 2.7.

2.6.3 HTTP

O *HTTP* (HyperText Markup Transfer Protocol) é o protocolo responsável pela transferência dos arquivos quando acessamos uma página *html*. Essa transferência ocorre através de portas, no caso do *HTTP* a porta é a 80. Por exemplo, um usuário que acessa uma pagina de jornal, ele digitará o endereço do site quando o navegador abre o site ele está transferindo os arquivos daquele site para o cache do usuário. É importante limparmos o

cache porque todas as vezes que acessamos um site os arquivos vão sendo acumulados no cache.

2.6.4 FTP

O *FTP* (File Transfer Protocol) é o protocolo responsável pela transferência de arquivos. Ele utiliza a porta 21. Um exemplo de *FTP* é o programa WS_FTP usado para transferir arquivos para o *WebGate*.

2.7 CRIAÇÃO DE PÁGINAS HTML

As páginas *HTML* podem ser desenvolvidas com qualquer editor *HTML (Front Page*, *DreamWaver*). Elas poderão conter os *applets*, programas desenvolvidos em linguagem Java, e/ou rotinas em Java script. Tanto as rotinas em Java quanto às em Java Script permitem o acesso a funções do CLP. Na Tabela 2.9 vemos um resumo da composição de uma página *html* com *xml* e *Java Script*

Tabela 2.9 - Estrutura de uma página *HTML* com xml e Java Script

***HTML* (cabeçalho da página)**
XML (declaração dos identificadores)
JAVASCRIPT (funções)
***HTML* (corpo da página)**

a) Identificadores XML

```
<XML id="style" src="xsl/exemplo.xsl"></XML>
<XML id="WriteOper"></XML>
<XML id="WPres1"></XML>
<XML id="WPres2"></XML>
<XML id="Temp"></XML>
<XML id="Velo"></XML>
<XML id="PTemp"></XML>
<XML id="PVelo"></XML>
<XML id="EstIlumi"></XML>
<XML id="EstArcon"></XML>
<XML id="EstMotor"></XML>
```

b) Funções Java Script

A linguagem Java Script tem como função escrever (tabela 2.10) e ler (tabela 2.11) o conteúdo de operandos através de comandos enviados ao CLP e exibidos na página *HTML*.

Tabela 2.10 – Escrita de operandos em Java Script

LINHA	JAVASCRIPT	Descrição da linha
1	<Script LANGUAGE="JavaScript">	Inicio do JavaScript
2	<!-- function writeOp(url,target) {	Inicio da função que recebe dois parâmetros, a URL e um numero identificador de alvo. Estes parâmetros são passados para a função na sua chamada.
3	WriteOper.async = false;	Torna falsa a propriedade async do browser. Com isso as tarefas são executadas uma de cada vez, sem tarefas concorrentes, para permitir a montagem do XML.
4	WriteOper.src = url;	Faz uma procura pela URL carregada em WriteOper.
5	WriteOper.load(url);	Carrega a URL.
6	if (target==1) { Target1.inner*HTML*=Write Oper.transformNode(style. XMLDocument);}	Se o parâmetro identificador for 1 a variável alvo será Target1 que será transformada segundo a folha de estilo definida no identificador style.
7	if (target==2) { Target2.inner*HTML*=Write Oper.transformNode(style.	Se o parâmetro identificador for 2 a variável alvo será Target2 que será transformada segundo a folha de
8	if (target==3){ Target3.inner*HTML*=Write Oper.transformNode(style. XMLDocument);}	Se o parâmetro identificador for 3 a variável alvo será Target3 que será transformada segundo a folha de estilo definida no identificador style.
9	} //--> </script>	Final da declaração do Script.

Tabela 2.11 –Leitura de operandos em Java Script

LINHA	JAVASCRIPT	DESCRIÇÃO DA LINHA
1	<SCRIPT FOR="window" EVENT="onload"> read(); </SCRIPT>	JavaScript que chama a função read(), no evento onload da página
2	function read() {	Início da função read()
3	Temp.async = false;	Torna falsa a propriedade async do browser, com isso as tarefas são executadas uma de cada vez sem tarefas concorrentes, para permitir a montagem do XML
4	TempURL ="ipp.xml?cmd=70&addr=m 1&format=D";	TempURL recebe a URL de comando de monitoração da memória 1 no formato decimal.
5	Temp.src= TempURL;	Faz uma procura pela URL carregada em TempURL.
6	Temp.load(TempURL);	Carrega a URL armazenada em WTempURL.
7	TempTarget.inner*HTML* = Temp.transformNode(style. XMLDocument);	Transforma o identificador XML Temp segundo sua folha de estilo e carrega no identificador de alvo TempTarget.

Para aumentar o número de monitorações devem ser duplicadas as linhas 3 à 7 dentro da função read(), modificando o nome de suas variáveis como TempURL, e utilizando-se outros identificador XML e alvo, conforme mostrado no exemplo a seguir.

Função original para monitoração da memória M0001 no formato decimal, utilizando o identificador <XML id="Temp"></XML>

```
Temp.async = false;
TempURL ="WebGate.xml?cmd=70&addr=m1&format=D";
Temp.src = TempURL;
Temp.load(TempURL);
TempTarget.innerHTML = Temp.transformNode(style.XMLDocument);
```

Nova função para monitorar a M0002 no formato decimal, utilizando o identificador <XML id="Velo"></XML>

```
Velo.async = false;
VeloURL ="WebGate.xml?cmd=70&addr=m2&format=D";
Velo.src = VeloURL;
Velo.load(VeloURL);
VeloTarget.innerHTML = Velo.transformNode(style.XMLDocument);
```

2.8 *WEBVIEW*

O *Webview* é um software composto por um banco de *applets* utilizados no desenvolvimento de páginas *HTML*.

Para a criação de telas *HTML* utilizando o *WebView* é necessário

- Biblioteca de *applets WebView*
- Plug-in Java JRE 1.4.2
- Um programa de *FTP*
- Um programa Editor *HTML*
- Um browser compatível

O *WebView* permite construir páginas *HTML* com as seguintes funcionalidades:

- Monitoração e apresentação no browser de valores analógicos e digitais de operandos do CLP;
- Envio de comandos digitais para o CLP através de botões típicos configuráveis (exemplo: botão Ligar, botão Desligar);
- Apresentação e edição de valores do CLP em controles típicos do Windows, tais como ComboBox, ListBox, CheckBox e RadioButton;
- Apresentação de alarmes e eventos a partir de valores de operandos do CLP;
- Atualização dinâmica de todos os *applets* da tela com uma freqüência configurável individualmente.

Na Tabela 2.12 estão os *applets* disponíveis para elaboração de sistemas:

Tabela 2.12– *Applets* Java disponíveis para implantação de aplicações

Driver	ColorAnimation	PenChart
AnalogInput	CheckBox	EventSummary
DigitalInput	RadioButton	AlarmSummary
AnalogOutput	ComboBox	Receita
DigitalOutput	ListBox	Variável Local
StringAnimation	TimeStamp	PushButton
ImageAnimation	BarGraph	

CRIANDO PÁGINAS COM *APPLET WEBVIEW*

No editor de páginas *HTML* deve-se:

Inserir o miniaplicativo Java;
Configurar as propriedades do miniaplicativo;
Informar o nome do Applet nas propriedades.

PARÂMETROS DOS *APPLETS*

Os *applets WebView* são configurados através de seus parâmetros (Tabela 2.13). Alguns parâmetros são obrigatórios enquanto outros são opcionais. Parâmetros obrigatórios precisam sempre ser configurados para que o applet execute corretamente. Os opcionais podem ser ou não configurados; caso não sejam, assumem um valor *default* correspondente.

Tabela 2.13 – Parâmetros de *applets* java

Operand OperTime OperAck	Precisa ser um operando válido do CLP
Background	Precisa ser um nome de cor ou um número hexadecimal ou um nome de arquivo válido
FontColor BarForeColor BarBackColor GraphColor AxisColor	precisa ser um nome de cor ou um número hexadecimal válido
Frequency FontSize DecimalDigits MinLimit MaxLimite NumInterval Value NumElem Div DivX DivY QtdValores NumEvents NumAlarms QtyPolling	Precisa ser um número inteiro válido

LIMITES DO *WEBVIEW*

Tabela 2.14– Limites do *WebView*

	Limites
Quantidade de *applets* por página *HTML*	Sem Driver: 10 *applets* Com Driver: 100 *applets* O Applet Driver é um concentrador de comunicações que visa otimizar a utilização do sistema
Quantidade e Tamanho de páginas *HTML*	Limite de memória do PO7091, PO9901 ou PO9900, tipicamente 150 kbytes. Nos PO7091 ou PO9901 pode-se Acrescentar expansão com via m card PO8541 com capacidade de 128Mb
Monitorações Cíclicas	Cinco *applets* a cada 1 segundo Dez *applets* a cada 2 segundos Com mais *applets*, ver limites do Driver
Limites do Driver: Capacidade	Até 16 blocos de operando com 64 operandos cada
Limites do Driver: Quantidade de Operandos Monitorados Simultaneamente	O limite é de 64 operandos por segundo, com um ou mais blocos, por exemplo: - Um bloco de 64 operandos a cada 1 segundo - Dois blocos de 32 operandos a cada 1 segundo - Três blocos de 42 operandos a cada 2 segundos - Quatro blocos de 48 operandos a cada 3 segundos - Cinco blocos de 64 operandos a cada 5 segundos
Menor período de monitoração	1 segundo
Maior período de monitoração	5 minutos

Desempenho do *WebView*

O desempenho do *WebView* em termos de tempo de resposta na apresentação de uma tela depende do 1) tempo de resposta do Servidor Web utilizado (PO7091, PO9901 ou PO9900) e também do 2) tempo da comunicação entre o CLP e Servidor Web. Os principais fatores que influenciam neste desempenho são mostrados na Tabela 2.15.

Tabela 2.15 – Desempenho

Tráfego na rede ethernet do Servidor Web	Em qualquer equipamento conectado a uma rede ethernet o desempenho poderá ser influenciado pelo tráfego na rede. Recomenda-se que o *WebView* seja utilizado em uma intranet ou internet com um tráfego em condições aceitáveis de utilização. Podem ser utilizados switches para isolar o tráfego em determinados segmentos para melhorar as condições gerais de utilização.
Número de Browsers abertos e Conexões simultâneas com o servidor	O desempenho diminui com a quantidade de conexões simultâneas abertas com o Servidor Web. Cada browser aberto representa no mínimo uma conexão. O applet Driver representa outra conexão. Cada applet de uma página *HTML* que não utiliza o Driver também representa mais uma conexão, sendo assim extremamente recomendável que se utiliza o applet Driver em páginas com mais de 5 *applets*.
Tempo de resposta do CLP	O tempo de resposta na comunicação entre o Servidor Web e o CLP é um dos principais fatores que limitam o desempenho do *WebView*. Cada uma das conexões simultâneas abertas precisa esperar que o CLP responda ao Servidor Web a solicitação feita pelo applet no browser. No caso do PO7091 a comunicação com o CLP é feita pelo barramento GBL, que é mais rápida do que no PO9901 ou PO9900, nos quais esta comunicação é feita pelo canal serial RS-232 do CLP.
Tempo de varredura do programa aplicativo do CLP	Cada solicitação feita pelo Servidor Web ao CLP é respondida somente no final do ciclo de varredura do programa aplicativo no CLP. Isto significa que programas aplicativos com um tempo de varredura maior levam mais tempo para responder e diminuem o desempenho geral do *WebView*. O tempo de varredura do programa aplicativo do CLP pode ser visualizado através do comando Informações no software de programação dos CLPs, MasterTool.
Tempo de carga do arquivo *webview.jar*	Quando um browser apresenta uma tela do *WebView*, o arquivo *webview.jar* é carregado do Servidor Web para o browser. Esta carga é relativamente lenta, tipicamente 3 ou 4 segundos, mas é feita uma única vez para cada browser aberto. Quando se troca de tela num mesmo browser, este arquivo não é recarregado. Recomenda-se que um sistema desenvolvido com o *WebView* utilize uma tela de abertura na qual aconteça a carga do *webview*.jar, para que durante a navegação nas demais telas do sistema este tempo de carga não se faça presente.

Mostramos os diversos tipos de tecnologias, modelos, características e limitações dos recursos que iremos utilizar em nosso trabalho. No capítulo 3 veremos algumas implementações que foram utilizadas usando essas tecnologias e a analise dos resultados obtidos.

CAPÍTULO 3

ANÁLISE E TESTES DE DESEMPENHO

Este capítulo tem como objetivo demonstrar os procedimentos utilizados nas práticas realizadas e resultados obtidos

- ✓ Acesso ao *WebGate* via cabo serial através do computador
- ✓ Alteração do numero IP do *WebGate*;
- ✓ Comunicação, através de cabo cruzado, entre o *WebGate* e o computador;
- ✓ Supervisão e alteração de variáveis do CLP via browser http;
- ✓ Acesso ao programa MásterTool através de configuração Ethernet (192.168.0.181);
- ✓ Monitoramento de variáveis pelo MasterTool.

3.1 CONFIGURANDO *WEBGATE* VIA HYPER TERMINAL

Antes de realizar a configuração devemos ligar um cabo serial PO8540 (cabo com dois DB9 fêmea em cada ponta).

A configuração via Hyper Terminal tem como objetivo fazer as alterações dos parâmetros de rede para que o *WebGate* seja acessado.

Para acesso via Hyper Terminal é necessário criar uma conexão através do programa HYPER TERMINAL do windows (Figura 3.1). Essa conexão é necessária para podermos fazer as configurações dos parâmetros de rede do *WebGate*.

Digitar o nome da conexão que será criada

Figura 3.1 - Nome da conexão

Escolher a porta pela qual será feito o acesso

Figura 3.2 – Porta que será acessada

Configurar parâmetros conforme listado abaixo

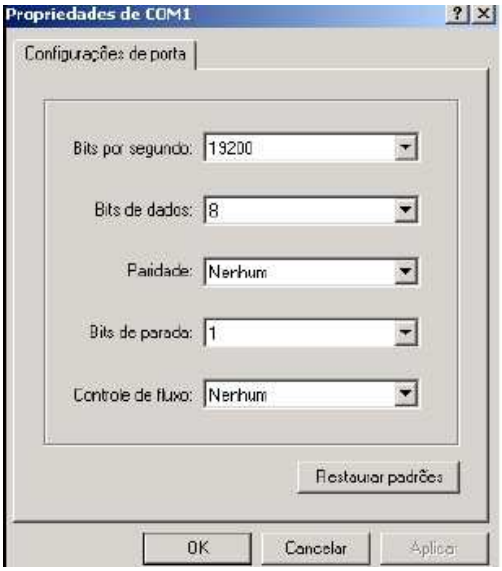


Figura 3.3 – Tela de parâmetros de configuração do Hiper Terminal

Ao acessar o *WebGate* ele mostra uma seqüência de inicialização com versão do Build, estado dos serviços (*FTP,WEB, TELNET*)

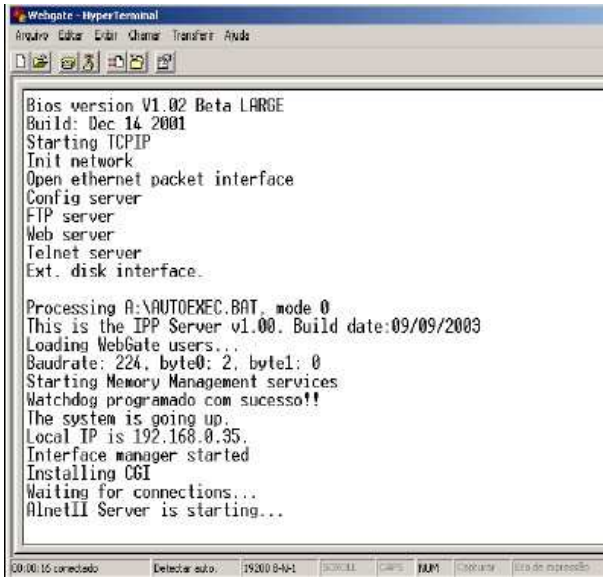


Figura 3.4 - Tela de inicialização do *WebGate* via Hyper Terminal

O comando ipcfg digitado no prompt de comando exibe as informações de endereço IP, máscara, *gateway*, *dhcp* e endereço *MAC* (Media Acess control).

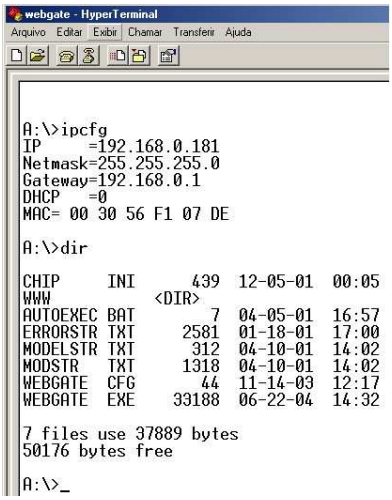


Figura 3.5 – Tela de visualização do diretório do *WebGate*

3.2 CONEXÃO DE EQUIPAMENTOS

A primeira parte a ser feita é a ligação entre os equipamentos com seus devidos cabos.

A Figura 3.6 mostra o esquema de conexão, os cabos e conectores.

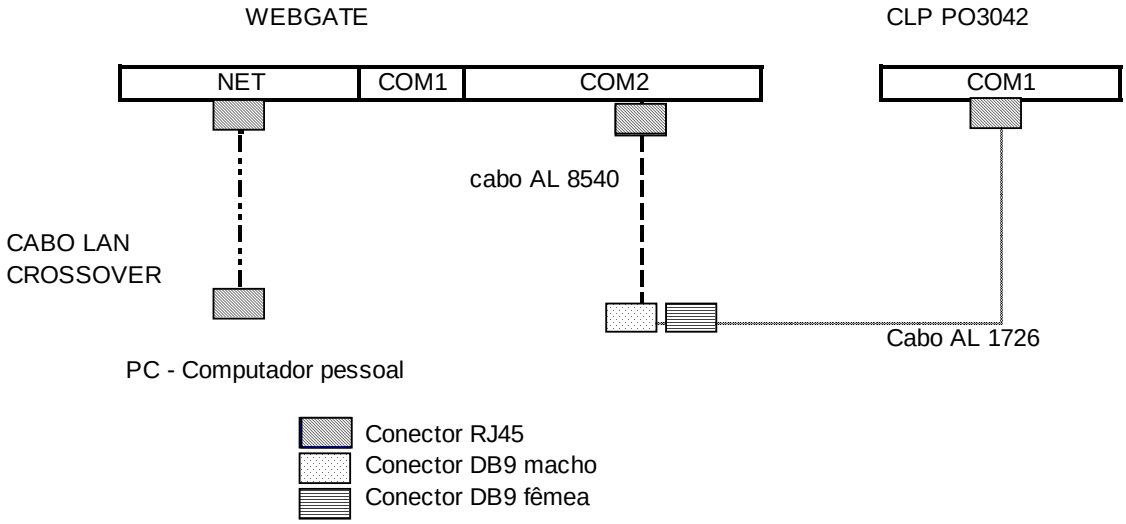


Figura 3.6- Conexão entre *WebGate* – PC e *WebGate* – CLP PO3042

Figura 3.7 - Bancada de simulações e testes com CLP PO3042

Devido a necessidade de ligação da IHM CIMREX 12 e do *WebGate*, tivemos que escolher outro modelo de UCP que tivesse duas entradas seriais (RS232). O modelo escolhido foi o PO3045, na Figura 3.8 ilustramos as conexões entre a UCP, *WebGate* e interface IHM CIMREX 12.

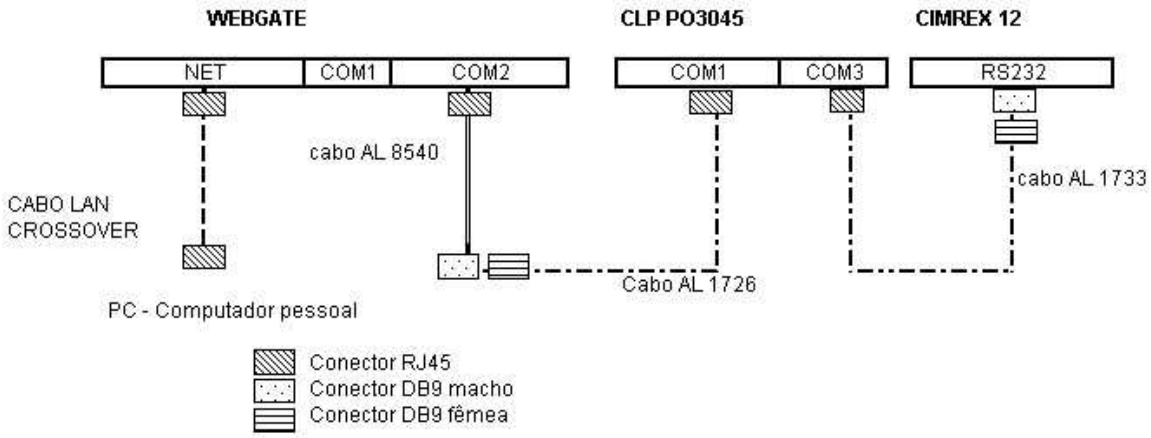


Figura 3.8 - Conexão entre *WebGate*–PC e *WebGate*–CLP PO3042–CIMREX 12

A Figura 3.9 apresenta a vista frontal da bancada vista na Figura 3.7

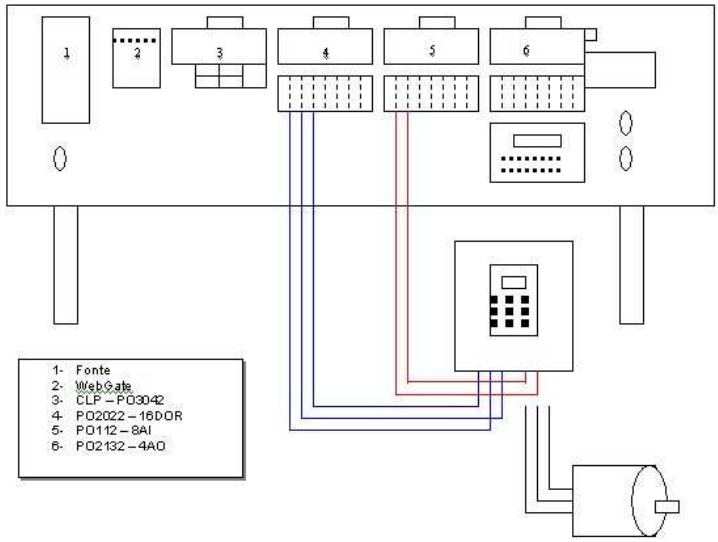


Figura 3.9– Diagrama da vista frontal da bancada vista

3.4 DESENVOLVENDO A PÁGINA HTML

A página *HTML* desenvolvida tem a função de supervisionar a planta. Podemos ver na Figura 3.10 a página *HTML* e na Figura 3.11 a planta industrial com os equipamentos montados.

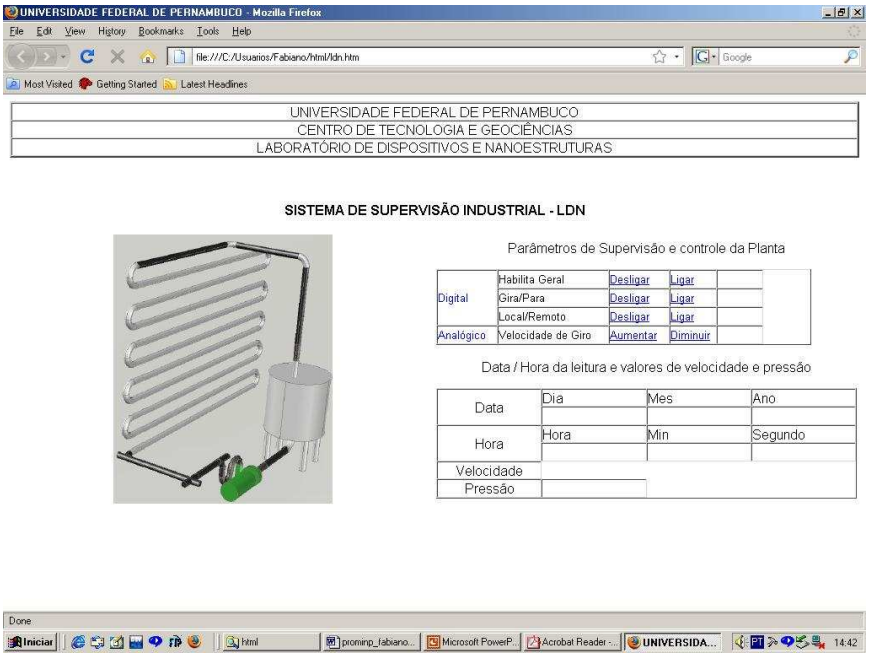


Figura 3.10 - Página html para supervisão da planta industrial

Figura 3.11 – Planta industrial com equipamentos montados

3.5 TRANSFERINDO ARQUIVOS PARA O *WEBGATE*

A transferência de arquivos é necessária para poder gravar os arquivos *HTML* e arquivos de configuração no *WebGate*. O procedimento para acesso e transferência e visto nas figuras 3.12 e 3.13.

ACESSANDO O WS FTP

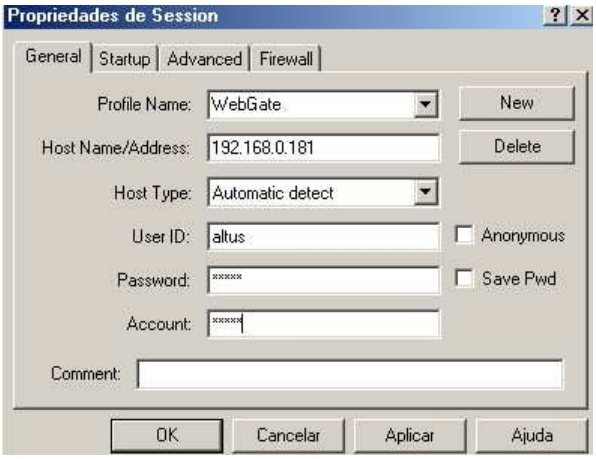


Figura 3.12 – Tela de acesso ao WS_FTP

Transferindo Arquivos

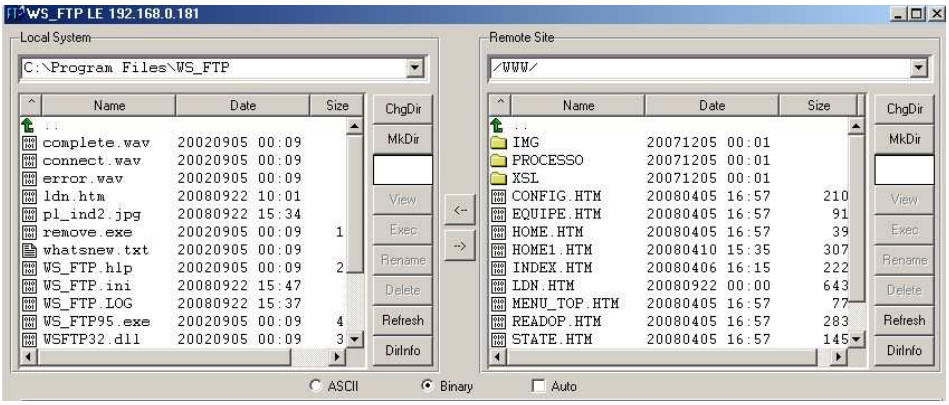


Figura 3.13 – Tela de transferência de arquivos do WS_FTP

3.6 TESTANDO A COMUNICAÇÃO E COMANDOS

Antes de visualizar as páginas *HTML*, deve-se testar a comunicação digitando no browser o comando 70. Outros comandos são disponíveis, conforme Tabela 3.1

http://192.168.0.181/webgate.xml?cmd=70&addr=m0001

Tabela 3.1 - Códigos de comando

Comando	Descrição
16	Escreve operandos
64	Status do equipamento
65	Status da comunicação
66	Status de forçamento
70	Leitura de operandos
71	Status do barramento de E/S
97	Diretório de módulos do programa

Fonte: Altus Sistemas de Informática S.A. Manual de Utilização do *WebGate* – PO9900. MU209690, 06/2003.

Como resultado obtido através do comando 64 digitado no browser temos as Figura 3.14 e Figura 3.15.

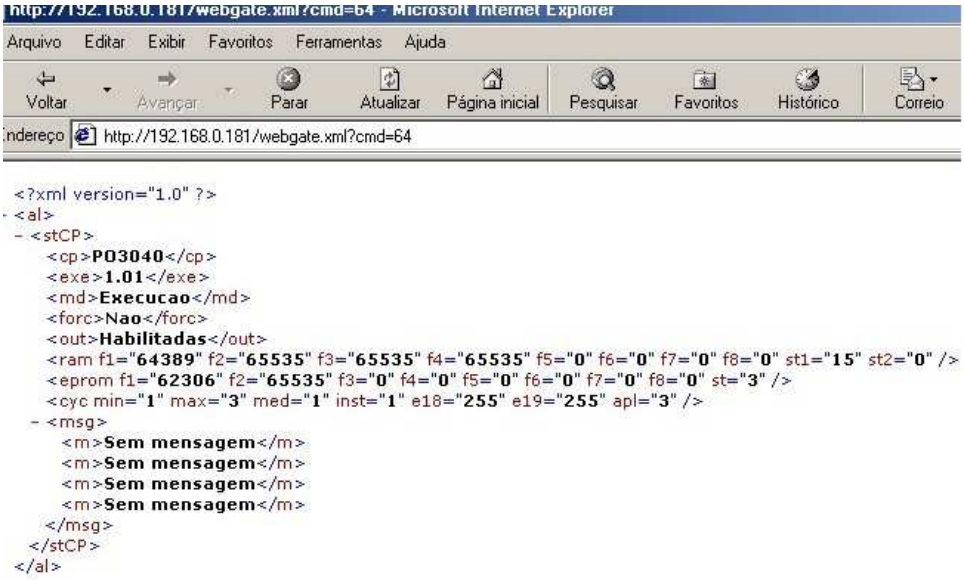


Figura 3.14 - Teste de comunicação via browser

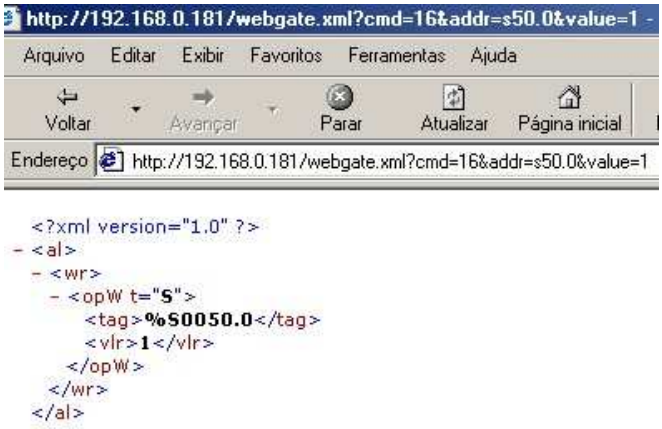


Figura 3.15 – Alteração do valor de variável com comando 16

3.7 MONITORAMENTO VIA MASTERTOOL

Para acessarmos o MasterTool através do *WebGate*, devemos informar o número IP do WebGate nas opções de endereço do MasterTool (Figura 3.16). Na Figura 3.17 temos o monitoramento das variáveis.

Figura 3.16 – Configuraçao do canal ethernet

Figura 3.17 – Monitoramento das variáveis %S0050.1 e %S0050.2

3.8 TEMPO DE RESPOSTA

Os tempos de resposta dependem do *WebGate*, da rede ethernet, do CLP e da conexão RS232.

- ✓ O tempo de uma requisição e resposta ALNET I no meio físico RS232 é de aproximadamente 1ms.
- ✓ O tempo que o CPL processa uma resposta ALNET I depende do tempo de varredura. Pode-se estimar em no máximo, 1 ciclo de varredura ou 50ms.

Teste feito com conexão ponto a ponto utilizando o comando PING -t. para pacotes com tamanho de 32 bytes tivemos os resultados apresentados na Tabela 3.2.

Tabela 3.2 – Tabela com pacotes enviados e recebidos

Pacotes enviados	2579
Pacotes recebidos	2579
Pacotes perdidos	0
Tempo mínimo	0
Tempo máximo	10ms

3.9 SIMULAÇÃO DE MEDIÇÕES

Os testes realizados para monitoramento de grandezas foram realizados com o modulo de entrada analógica PO1112. As grandezas lidas pelo módulo foram tensão, corrente, resistência. A geração de tensão foi feita de duas formas: com fonte de tensão e gerador de onda. A Figura 3.18 mostra a estrutura com CLP PO3045 utilizada na simulação. A geração de corrente foi através de fonte de corrente Figura 3.19 e a geração de resistência foi com um potenciômetro.

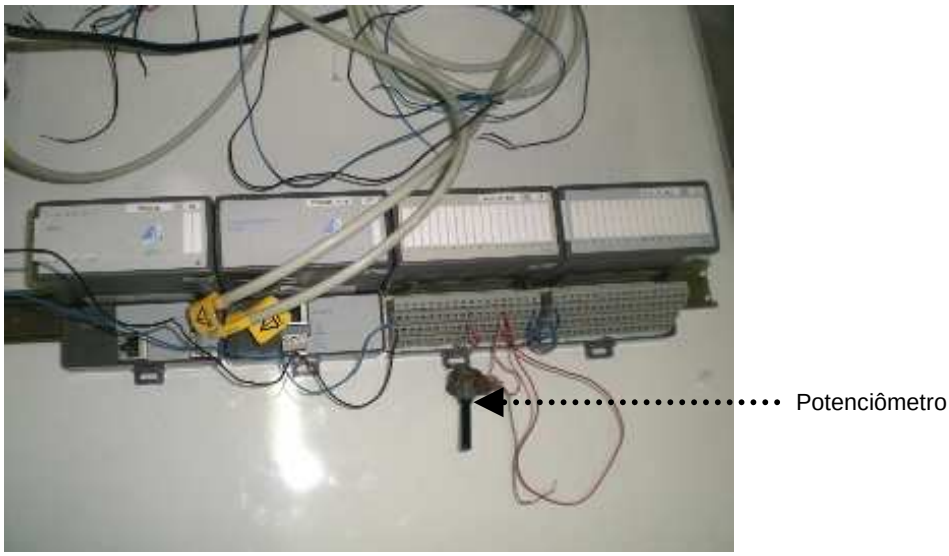


Figura 3.18 - Estrutura utilizada na simulação

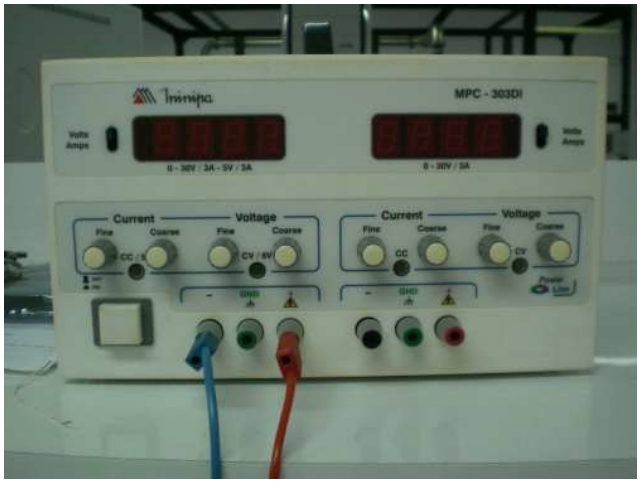


Figura 3.19 - Fonte de tensão e corrente

Figura 3.20 - Gerador de sinal

Para medição de tensão devemos ligar a saída positiva do dispositivo gerador de tensão (sensor, fonte, etc) a entrada L(0,1,2,...,7) e a saída negativa do dispositivo gerador de tensão na entrada N(0,1,2,...7). Como pode ser visto no diagrama esquemático do módulo PO1112 [APÊNDICE A], ele possui 8 canais variando de 0 a 7. A escolha de um desses canais dependerá da configuração feita no PROPONTO. Em nossa simulação fizemos a configuração do canal 0.

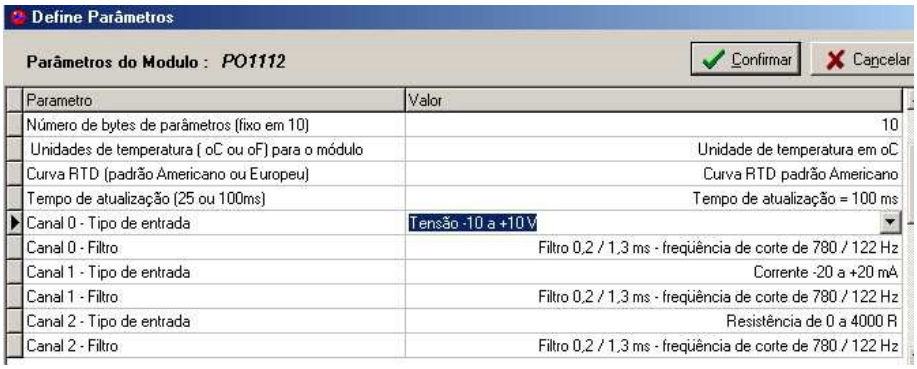

Define Parâmetros
Parâmetros do Modulo : PO1112
Confirmar
Cancelar
Parametro
Valor
Número de bytes de parâmetros (fixo em 10)
10
Unidades de temperatura (oC ou oF) para o módulo
Unidade de temperatura em oC
Curva RTD (padrão Americano ou Europeu)
Curva RTD padrão Americano
Tempo de atualização (25 ou 100ms)
Tempo de atualização = 100 ms
Canal 0 - Tipo de entrada
Tensão -10 a +10 V
Canal 0 - Filtro
Filtro 0,2 / 1,3 ms - freqüência de corte de 780 / 122 Hz
Canal 1 - Tipo de entrada
Corrente -20 a +20 mA
Canal 1 - Filtro
Filtro 0,2 / 1,3 ms - freqüência de corte de 780 / 122 Hz
Canal 2 - Tipo de entrada
Resistência de 0 a 4000 R
Canal 2 - Filtro
Filtro 0,2 / 1,3 ms - freqüência de corte de 780 / 122 Hz

Figura 3.21 – Configuração de canais do PO1112

Como a configuração foi feita no canal 0, a ligação será em L0 e N0. Para monitoramento e tratamento em linguagem *Ladder* no MásterTool a variável utilizada foi a %M0200.

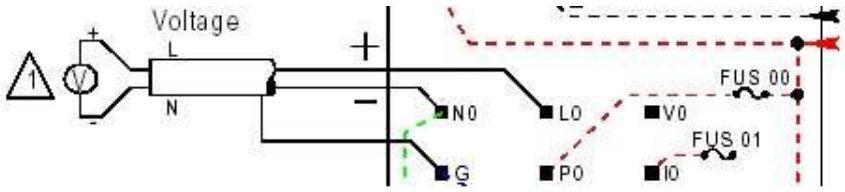


Figura 3.22 – Diagrama de ligação de tensão

A medição de corrente foi realizada no canal 1, ou seja, o pólo positivo da fonte foi ligado ao borne I1 e o pólo negativo da fonte ao borne N1. Para monitoramento e tratamento em linguagem *LADDER* no MásterTool a variável utilizada foi a %M0201.

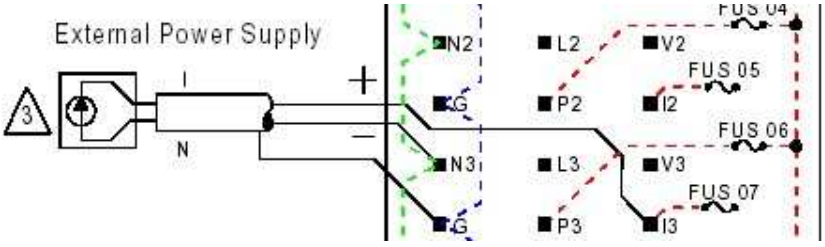


Figura 3.23 – Diagrama de ligação de fonte de corrente

A medição de resistência foi realizada no canal 2. Fizemos a simulação com resistência e potenciômetro. Ligamos as extremidades das pernas do potenciômetro e do resistor a V2 e N2, L2 e V2. Para monitoramento e tratamento em linguagem *Ladder* no MásterTool a variável utilizada foi a %M0202.

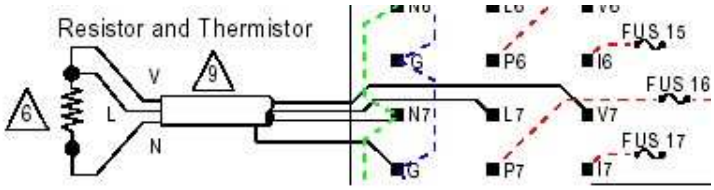


Figura 3.24 – Diagrama de ligação de resistor e termistor

3.10 GRÁFICOS DE RESULTADO DA SUPERVISÃO VIA PÁGINA *HTML*

A simulação da medição de tensão foi feita através de um gerador de onda PM 5190 LF SinThesizer 2mHz – 2MHz Figura 3.20.

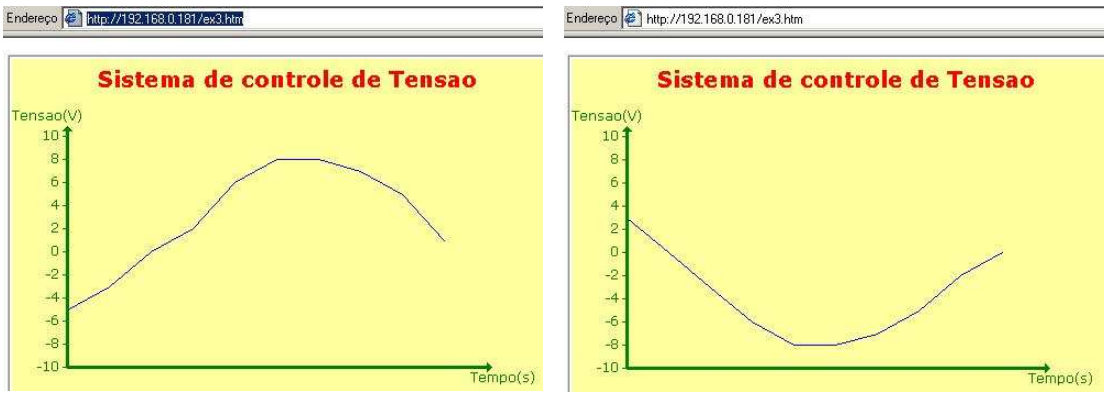


Figura 3.25 – Onda senoidal com frequência de 20Hz – 16Vpp

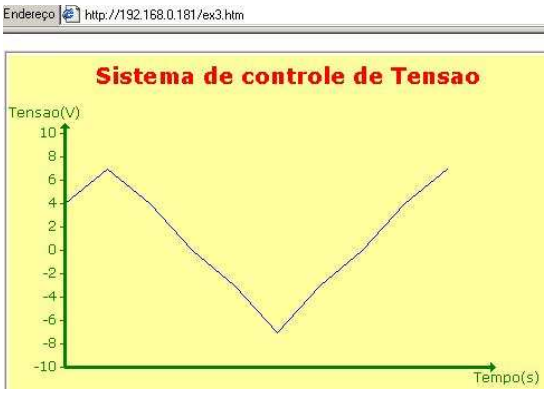


Figura 3.26 - Onda triangular
Freqüência de 20Hz – 16 Vpp

Figura 3.27 - Onda quadrada Freqüência 20Hz – 16Vpp

Podemos perceber pela figura 3.28 que as formas de onda já começam a sofre uma alteração bastante significativa quando aumentamos a freqüência para 50Hz.

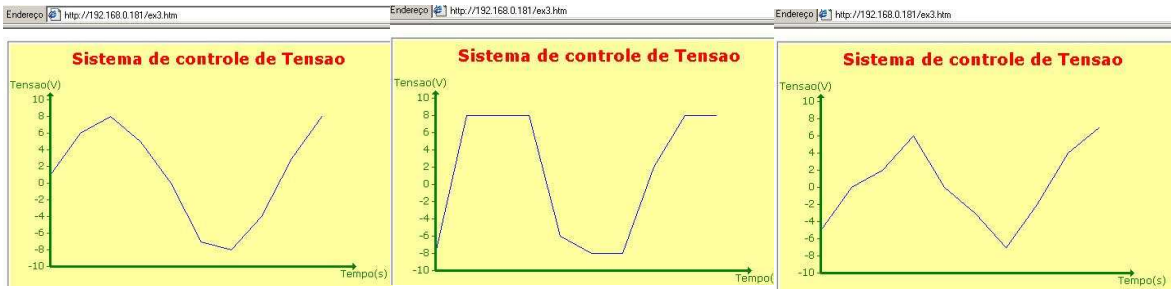


Figura 3.28 – Formas de onda senoidal, quadrada triangular para 50Hz 16Vpp

Com o aumento da freqüência para 100Hz as formas de onda ficam totalmente distorcidas, conforme pode ser visto na figura 3.29.

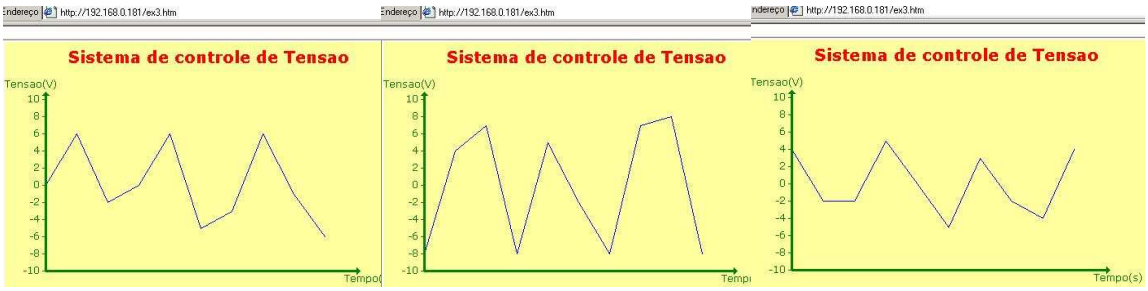


Figura 3.29 - Formas de onda senoidal, quadrada triangular para 100Hz 16Vpp

3.11 ARMAZENAMENTO EM TABELAS

A primeira etapa para utilização de tabelas é a configuração que é feita no arquivo c-cont.000, conforme mostrada na Figura 3.30.

Tabela utilizada TM000

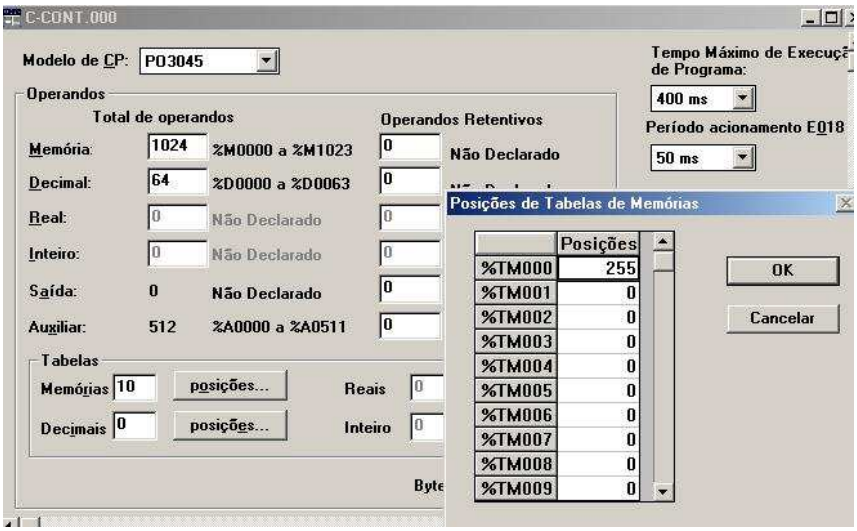


Figura 3.30 - Configuração da tabelas

MONITORAMENTO

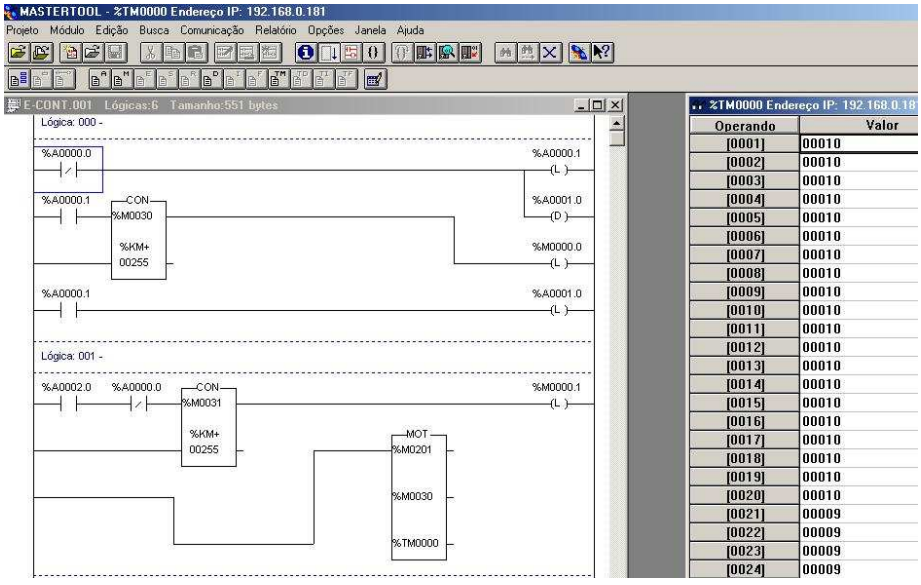
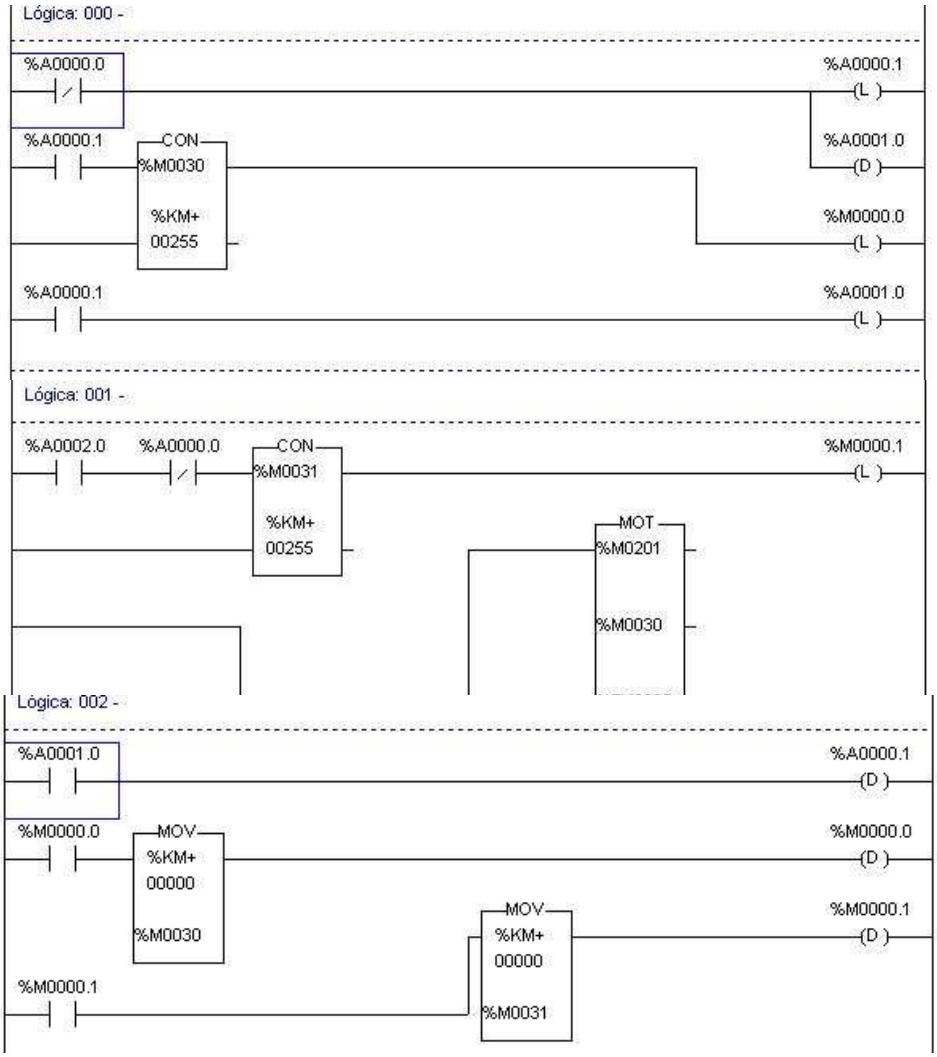
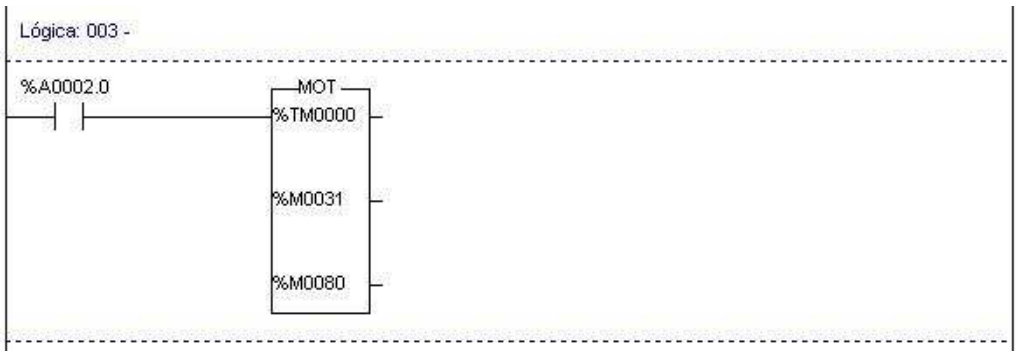


Figura 3.31 – Monitoramento de operandos tabela

As lógicas abaixo foram desenvolvidas para armazenar o sinal gerado através do gerador de ondas em uma tabela e exibir esse conteúdo logo após um tempo determinado por um temporizador.

Foi necessário desenvolver esse rotina porque estava havendo perda de alguns pontos, conforme aumentássemos o valor da freqüência.





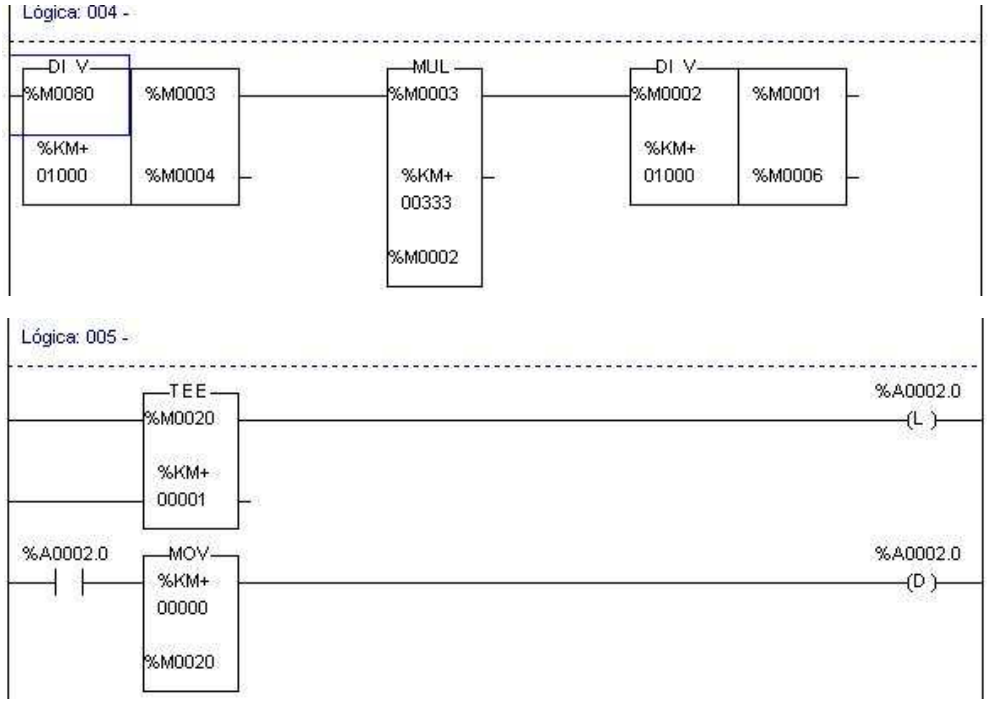


Figura 3.32 – Lógicas para gravação e leitura de dados numa tabela

Conforme foi visto neste capítulo, conseguimos realizar com sucesso a configuração do endereço *IP* do Webgate para que o mesmo fosse acessado pela rede. O meio de comunicação entre o *WebGate* e o computador foi um cabo *crossover*.

Alem disso, também fizemos configuração do MasterTool para acesso remoto. Com isso passamos a desenvolver as rotinas usando a conexão através da rede.

Utilizamos o gráfico *Penchart* do *Webview*, porém os resultados obtidos nas formas de onda não foram adequados. Tentamos resolver esse problema utilizando o recurso de Tabelas que possui uma capacidade para 255 valores, mas também não obtivemos resultado por causa da rapidez na gravação dos campos da tabela. A solução para esse problema será fruto para futuras análises

CAPÍTULO 4

CONCLUSÕES E TRABALHOS FUTUROS

O trabalho teve como objetivo inicial permitir que o processo industrial se comunique a longa distância, ou seja, realizar de qualquer local que esteja em rede a supervisão da planta no que se refere ao funcionamento e monitoramento. O meio escolhido para esse compartilhamento foi a *intranet/internet* devido a popularização e as ferramentas disponíveis.

A Automação industrial e a Instrumentação vêm evoluindo em parceria com as novas tecnologias. Isso ficou muito claro no trabalho por causa da capacidade de resultado e a forma de apresentação dos mesmos. Como base de tudo foi utilizada a tecnologia *HTML* e J*ava*. Aliadas a essas duas linguagens estão o XML e XSL que permitiram formatação dos resultados obtidos através dos navegadores.

A apresentação de resultados através de gráficos é muito mais fácil para o entendimento. Foi por isso que utilizamos um recurso gráfico em linguagem J*ava* chamado *Penchart*.

As ferramentas que foram utilizadas para desenvolver o trabalho foram: *Webgate*, UCP PO3045, *WevView*, PO1112 todos da Altus Sistemas de Informática. Contudo, outros fabricantes, tecnologias e modelos podem ser utilizados.

Iniciamos o trabalho com a PO3042, mas depois percebemos que ele não iria atender a necessidade de compartilhar e permitir o uso de IHM. Sendo assim, trocamos para o P03045. Esse tipo de situação é comum acontecer e demonstra a flexibilidade de troca por outro modelo.

As linguagens *HTML, Java Script e Ladder* são fáceis para implementar algumas coisas e difíceis para implementar outras.

As limitações e dificuldades apresentadas pelas linguagens foram alguns dos obstáculos enfrentados durante o trabalho. O programa MasterTool 4100 só realiza programação em *ladder*, mas existe a versão MT 8000 que oferece opção em linguagem ST. O modelo S7 200 da Simmens oferece as duas opções *ladder* e ST ao mesmo tempo.

Também encontramos limitações nos tipos de operandos utilizados no modelo PO3045. Tivemos a necessidade de usar ponto flutuante, mas o mesmo só trabalha com memórias, auxiliar e decimal. O *applet Penchart* também apresentou limitações ao exibir sinais com freqüências acima de 20Hz. Acima desse valor os sinais passam a apresentar pequenas perdas, ou seja, o *applet* não consegue fazer uma amostragem em tempo real porque o período de atualização é um parâmetro que só aceita valores inteiros e cujo tempo mínimo é de 1 segundo. Tentamos resolver esse problema desenvolvendo uma rotina em *ladder* utilizando tabelas, mas não foi suficiente porque um operando tabela tem capacidade

para 255 posições e como o tempo de atualização é muito rápido, os valores são substituídos antes de serem exibidos no gráfico. Tentamos utilizar um temporizador, porém pelo fato de usar o PO3045 o tempo mínimo de temporização é de 0,1 segundo.

O *WebGate* PO9900 possui o limite de 150Kbytes de espaço disponível para aplicações. Esse espaço limita muito a quantidade de aplicações e os recursos utilizados para o seu desenvolvimento. Para aumentar o espaço do *WebGate* é necessário fazer a troca por outro modelo.

O objetivo principal foi atingido, disponibilizar acesso para fazer a supervisão remota. Algumas limitações foram detectadas e poderão ser fruto de trabalhos futuros na melhoria desse sistema.

Assim como os meios de comunicação estão evoluindo, as tecnologias para desenvolvimento também crescem juntas. Um novo sistema supervisório poderia ser desenvolvido com I*nTouch*, *Eclipse Java* ou *LabView*. Todos são sistemas que oferecem uma boa variedade de recursos e algumas limitações que foram encontradas com *HTML* e *java* poderiam ser solucionadas por um deles. Outra coisa bastante importante seria o armazenamento das informações coletadas. Poderia ser feito um banco de dados com todos os resultados obtidos durante a supervisão remota.

APÊNDICE A

Este apêndice fornece procedimento passo a passo para configuração de módulos via PROPONTO, diagramas internos dos módulos, código fonte dos sistemas desenvolvidos.

A-1 CONFIGURANDO PROPONTO

Clicar no botão **BARRAMENTO**

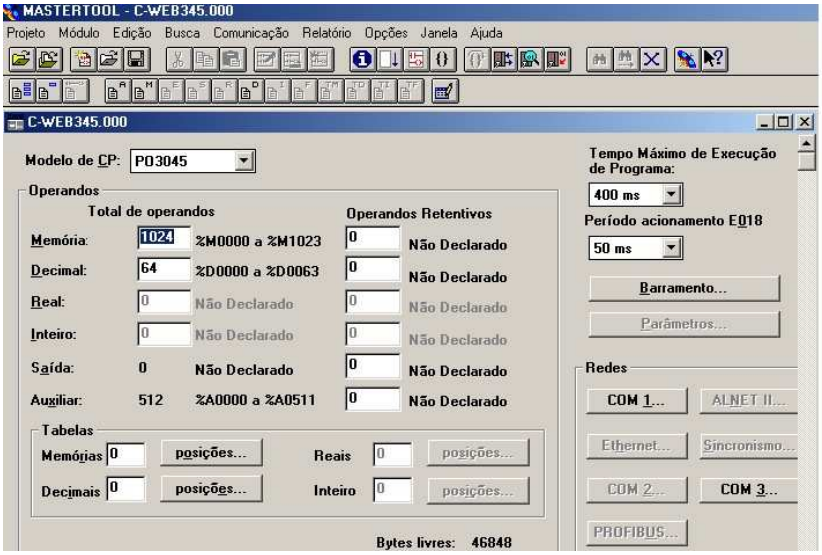

MASTERTOOL - C-WEB345.000
Projeto Módulo Edição Busca Comunicação Relatório Opções Janela Ajuda
C-WEB345.000
Modelo de CP: PO3045
Tempo Máximo de Execução de Programa:
400 ms
Período acionamento E018
50 ms
Barramento...
Parâmetros...
Operandos
Total de operandos
Operandos Retentivos
Memória: 1024 %M0000 a %M1023 0 Não Declarado
Decimal: 64 %D0000 a %D0063 0 Não Declarado
Real: 0 Não Declarado 0 Não Declarado
Inteiro: 0 Não Declarado 0 Não Declarado
Saída: 0 Não Declarado 0 Não Declarado
Auxiliar: 512 %A0000 a %A0511 0 Não Declarado
Tabelas
Memórias 0 posições... Reais 0 posições...
Decimais 0 posições... Inteiro 0 posições...
Bytes livres: 46848
Redes
COM 1... ALNET II...
Ethernet... Sincronismo...
COM 2... COM 3...
PROFIBUS...

Figura A.1– Tela inicial de configuração do barramento

Escolher o **Segmento 0**

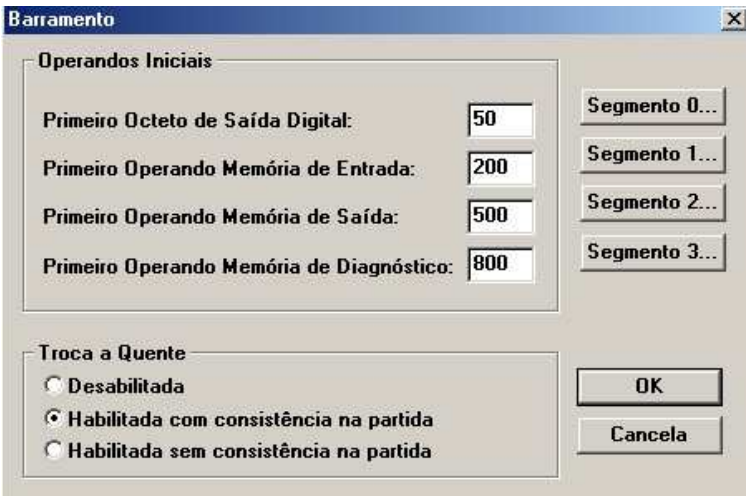


Figura A.2 - Tela de configuração dos segmentos

A tela exibe os módulos, entradas e saídas que estão salvos na memória atual do CLP.

CLICAR no botão **EXECUTAR**

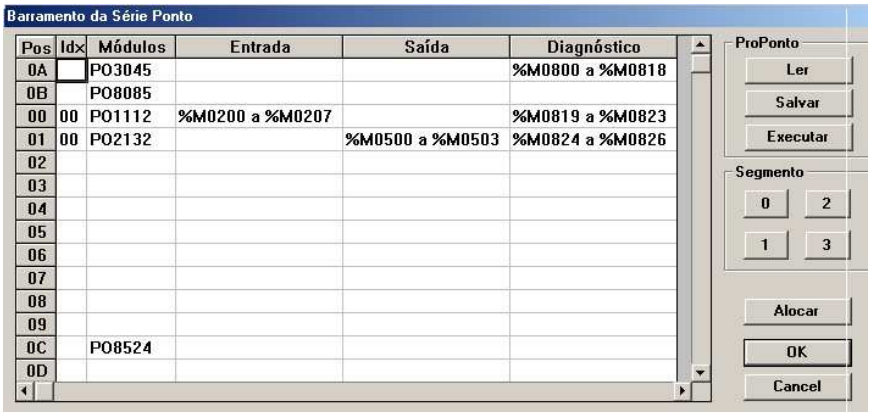


Figura A.3 – Tela com especificações do barramento

O sistema irá executar o programa **PROPONTO.** Nele adicionamos no segmento 0 a UCP que estamos trabalhando, os módulos de entrada e saída e as bases.

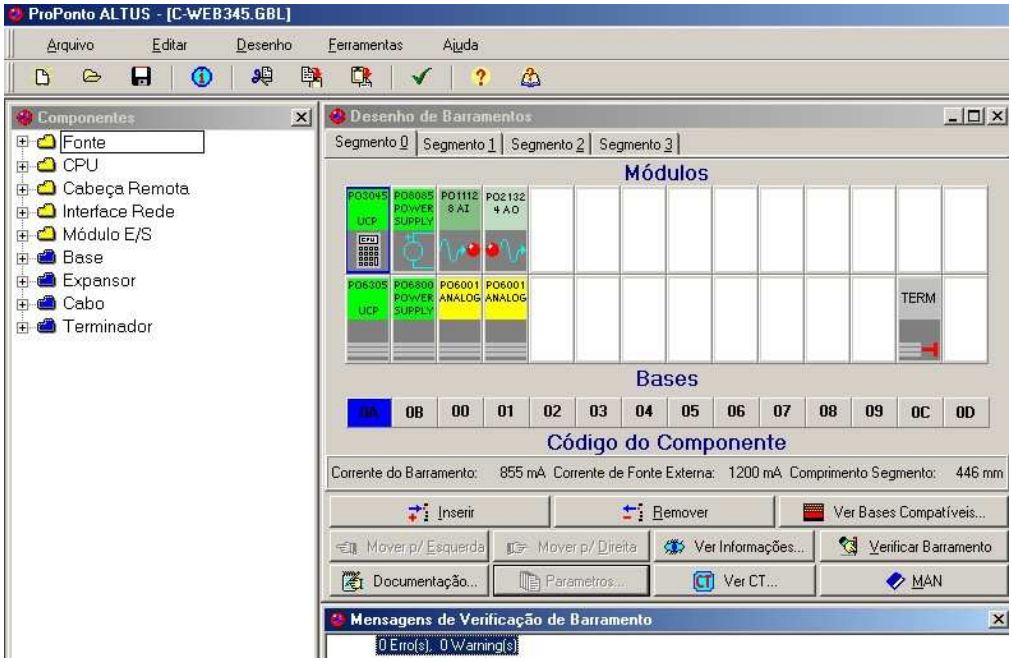


Figura A.4 – Tela do proponto

Ao clicar com o botão direito sobre a base PO112. Escolhemos a opção parâmetros. Devemos configurar os canais conforme a entrada que está sendo medida (tensão, corrente, resistência).

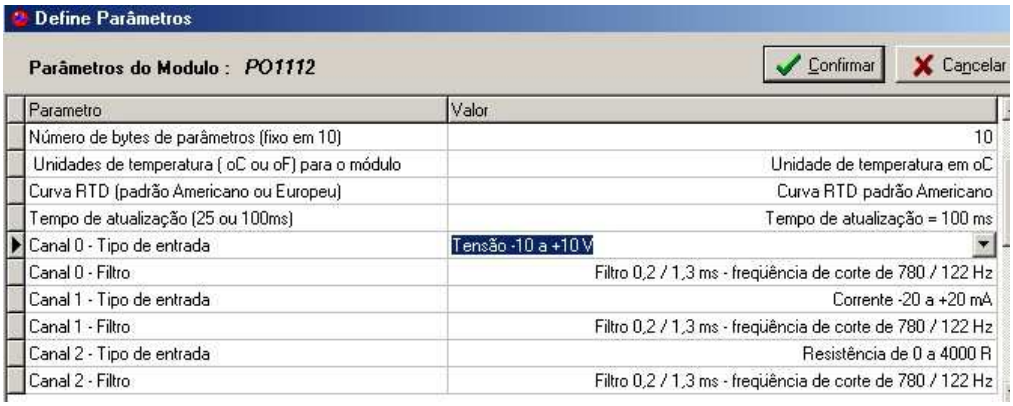

Define Parâmetros
Parâmetros do Modulo : PO1112
Confirmar
Cancelar
Parametro
Valor
Número de bytes de parâmetros (fixo em 10)
10
Unidades de temperatura (oC ou oF) para o módulo
Unidade de temperatura em oC
Curva RTD (padrão Americano ou Europeu)
Curva RTD padrão Americano
Tempo de atualização (25 ou 100ms)
Tempo de atualização = 100 ms
Canal 0 - Tipo de entrada
Tensão -10 a +10 V
Canal 0 - Filtro
Filtro 0,2 / 1,3 ms - freqüência de corte de 780 / 122 Hz
Canal 1 - Tipo de entrada
Corrente -20 a +20 mA
Canal 1 - Filtro
Filtro 0,2 / 1,3 ms - freqüência de corte de 780 / 122 Hz
Canal 2 - Tipo de entrada
Resistência de 0 a 4000 R
Canal 2 - Filtro
Filtro 0,2 / 1,3 ms - freqüência de corte de 780 / 122 Hz

Figura A.5 - Parâmetros de configuração da base de entrada PO1112

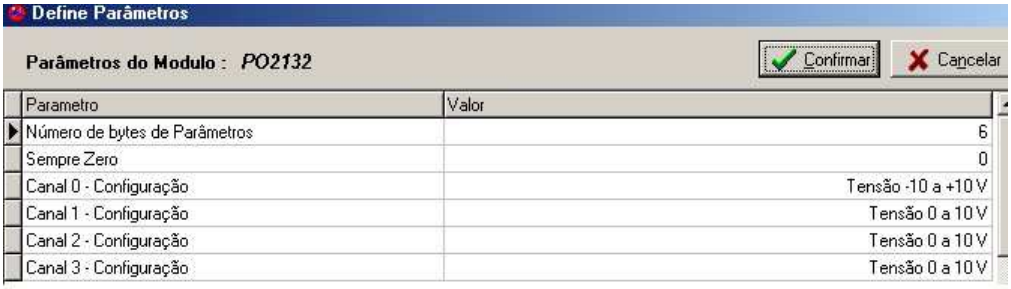


Figura A.6 - Parâmetros de configuração da base de saída PO2132

Depois de retornar da tela do proponto, devemos ler e salvar a configuração. Aparecerá uma tela perguntando se deseja substituir os módulos. Devemos escolher SIM.

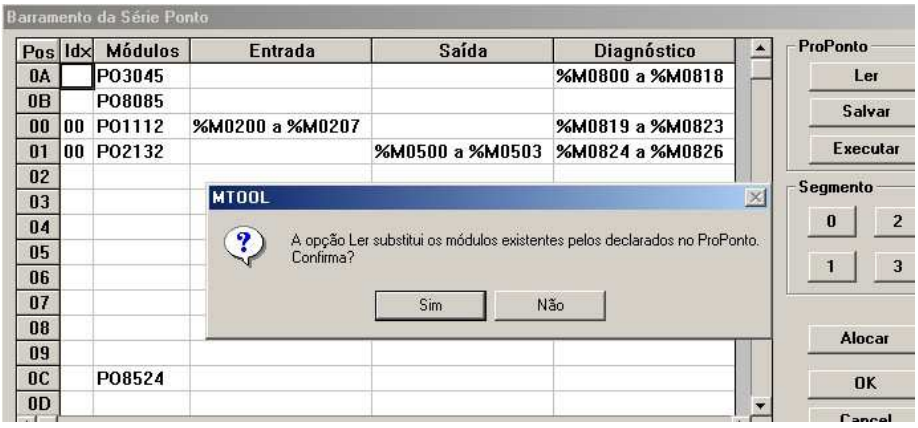


Figura A.7 - Tela para salvar as configurações feitas no proponto

A-2 CÓDIGO FONTE DA PÁGINA *HTML*

```
<HTML>
<head>
<meta http-equiv="Content-Language" content="pt-br">
<meta http-equiv="Content-Type" content="text/HTML; charset=windows-1252">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>UNIVERSIDADE FEDERAL DE PERNAMBUCO</title>
</head>

<body>
<table border="2" width="100%" cellspacing="0" cellpadding="0">
  <tr>
    <td width="100%" align="center"><font face="Arial">UNIVERSIDADE FEDERAL DE
      PERNAMBUCO</font></td>
  </tr>
  <tr>
    <td width="100%" align="center"><font face="Arial">CENTRO DE TECNOLOGIA E
      GEOCIÊNCIAS</font></td></tr>
  <tr>
    <td width="100%" align="center"><font face="Arial">LABORATÓRIO DE
      DISPOSITIVOS E NANOESTRUTURAS</font></td>
  </tr> </table><p> </p>

<p align="center"><b><font face="Arial">SISTEMA DE SUPERVISÃO INDUSTRIAL -
LDN</font></b></p>

<XML id="style" src="/xsl/model.xsl"></XML>
<XML id="WriteOper"></XML>
<XML id="WPres1"></XML>
<XML id="WPres2"></XML>
<XML id="Temp"></XML>
<XML id="Velo"></XML>
<XML id="PTemp"></XML>
<XML id="PVelo"></XML>
<XML id="EstIlumi"></XML>
<XML id="EstArcon"></XML>

<XML id="LocalRemoto"></XML>
<XML id="HabGeral"></XML>
<XML id="GiraPara"></XML>
<XMLS id="VeloGiro"></XML>

 <script LANGUAGE="JavaScript">

 <!--
    function writeOp(url,target) {
   WriteOper.async = false;
    WriteOper.src = url;
    WriteOper.load(url);
   if (target==1) {Target1.innerHTML =
WriteOper.transformNode(style.XMLDocument);}
   if (target==2) {Target2.innerHTML =
WriteOper.transformNode(style.XMLDocument);}
   if (target==3) {Target3.innerHTML =
WriteOper.transformNode(style.XMLDocument);}
   if (target==4) {Target4.innerHTML =
WriteOper.transformNode(style.XMLDocument);}

    }
```

```
   function writePes(id){
    if (id == 1) {
    WURL = "/WebGate.xml?cmd=16&addr=m50&value="+PresTemp.value;
    WPres1.async = false;
    WPres1.src = WURL;
    WPres1.load(WURL);
    TargetPres1.innerHTML = WPres1.transformNode(style.XMLDocument);
    }
    if (id == 2) {
    WURL = "/WebGate.xml?cmd=16&addr=m51&value="+PresVelo.value;
    WPres2.async = false;
    WPres2.src = WURL;
     WPres2.load(WURL);
    TargetPres2.innerHTML = WPres2.transformNode(style.XMLDocument);
    }
}
    function read() {

    HabGeral.async = false;
    HabGeralURL ="/WebGate.xml?cmd=70&addr=m050&format=D";
    HabGeral.src = HabGeralURL;
    HabGeral.load(HabGeralURL);
    HabGeralTarget.innerHTML = HabGeral.transformNode(style.XMLDocument);

    GiraPara.async = false;
    GiraParaURL ="/WebGate.xml?cmd=70&addr=A1.2&format=D";
    GiraPara.src = GiraParaURL;
    GiraPara.load(GiraParaURL);
    GiraParaTarget.innerHTML = GiraPara.transformNode(style.XMLDocument);

    LocalRemoto.async = false;
    LocalRemotoURL ="/WebGate.xml?cmd=70&addr=A1.3&format=D";
    LocalRemoto.src = LocalRemotoURL;
    LocalRemoto.load(LocalRemotoURL);
    LocalRemotoTarget.innerHTML = LocalRemoto.transformNode(style.XMLDocument);
   }
  //-->
  </script>
<SCRIPT FOR="window" EVENT="onload">
   read();
</SCRIPT>
<table border="0" width="100%">
  <tr>
    <td width="50%" align="center">
      <p align="center"><img border="0" src="img/pl_ind2.jpg" width="260"
height="319" align="middle" alt="Planta Industrial - LDN"></p>
    </td>
    <td width="50%">
      <p align="center"><font face="Arial">Parâmetros de Supervisão e controle
      da Planta</font></p>
      <table border="1" width="445" cellspacing="0" cellpadding="0">
        <tr>
          <td width="70" rowspan="3"><font face="Arial" color="#0000FF"
size="2">Digital</font></td>
          <td width="130"><font face="Arial" size="2">Habilita Geral</font></td>

          <td width="70"><a
href="javascript:writeOp('/WebGate.xml?cmd=16&amp;addr=e0.0&amp;value=0',1)"><fon
t face="Arial" size="2">Desligar</font></a></td>
          <td width="54"><a
href="javascript:writeOp('/WebGate.xml?cmd=16&amp;addr=e0.0&amp;value=1',1)"><fon
t face="Arial" size="2">Ligar</font></a></td>
```

```
          <td width="53"><div align="center" id="Target1"> </div></td>
          <td width="54"><div id="HabGeralTarget"></div> </td>
        </tr><tr>

          <td width="130"><font face="Arial" size="2">Gira/Para</font></td>
          <td width="70"><a
href="javascript:writeOp('/WebGate.xml?cmd=16&amp;addr=a1.1&amp;value=0',2)"><fon
t face="Arial" size="2">Desligar</font></a></td>
          <td width="54"><a
href="javascript:writeOp('/WebGate.xml?cmd=16&amp;addr=a1.1&amp;value=1',2)"><fon
t face="Arial" size="2">Ligar</font></a></td>
          <td width="53"><div align="center" id="Target2"> </div></td>
          <td width="54"><div id="GiraParaTarget"></div> </td></tr>
        <tr>
          <td width="130"><font face="Arial" size="2">Local/Remoto</font></td>
          <td width="70"><a
href="javascript:writeOp('/WebGate.xml?cmd=16&amp;addr=a1.2&amp;value=0',3)">
<font face="Arial" size="2">Desligar</font></a></td>
          <td width="54"><a
href="javascript:writeOp('/WebGate.xml?cmd=16&amp;addr=a1.2&amp;value=1',3)"><fon
t face="Arial" size="2">Ligar</font></a></td>
          <td width="53"><div align="center" id="Target3"> </div></td>
          <td width="54"><div id="LocalRemotoTarget"> </div> </td></tr>
        <tr>
          <td width="70"><font face="Arial" size="2">Analógico</font></td>
          <td width="130"><font face="Arial" size="2">Velocidade de
Giro</font></td>
          <td width="70"><font size="2" face="Arial">Aumentar</font></td>
          <td width="54"><font size="2" face="Arial">Diminuir</font></td>
          <td width="53"> </td></tr>
      </table>
      <p align="center"><font face="Arial">Data / Hora da leitura do
sensor</font></p>
      <table border="1" width="100%" cellspacing="0" cellpadding="0">
        <tr>
          <td width="25%" rowspan="2">
            <p align="center"><font face="Arial">Data</font></td>
          <td width="25%"><font face="Arial">Dia</font></td>
          <td width="25%"><font face="Arial">Mes</font></td>
          <td width="25%"><font face="Arial">Ano</font></td>
        </tr>
        <tr>
          <td width="25%"> </td><td width="25%"> </td>
          <td width="25%"> </td></tr>
        <tr>
          <td width="25%" rowspan="2" align="center"><font
face="Arial">Hora</font></td>
          <td width="25%"><font face="Arial">Hora</font></td>
          <td width="25%"><font face="Arial">Min</font></td>
          <td width="25%"><font face="Arial">Segundo</font></td>
        </tr>
        <tr>
          <td width="25%"> </td><td width="25%"> </td>
          <td width="25%"> </td></tr>
      </table></td></tr></table>
</body>
</HTML>
```

A-3 Código fonte de estilo

```
<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/TR/WD-xsl">
<xsl:template match="text()">
<xsl:value-of />
</xsl:template>

<xsl:template match="*">
<xsl:apply-templates />
</xsl:template>
<xsl:template match="/"> <xsl:apply-templates select="//al" /> </xsl:template>

<xsl:template match="rd"> <xsl:apply-templates select="opR"/> </xsl:template>

<xsl:template match="wr">
        <xsl:apply-templates select="opW"/>
        <xsl:apply-templates select="err"/>
</xsl:template>

<xsl:template match="opW">
                <xsl:if match=".[@t='A']">
                        <xsl:if match=".[vlr='1']">
                   <font face="Arial Narrow" color="#000000">Lig Exec</font>
                        </xsl:if>
                        <xsl:if match=".[vlr='0']">
            <font face="Arial Narrow" color="#000000">Des Exec</font>
                        </xsl:if>
                </xsl:if>

                <xsl:if match=".[@t='M']">
                  <font face="Arial Narrow" color="#000000"> <xsl:value-of select="vlr"/></font>
          </xsl:if>
</xsl:template>

<xsl:template match="opR">

        <xsl:if match=".[@t='M']">
                <font face="Arial Narrow" color="#000000"> <xsl:value-of select="vlr"/></font>
        </xsl:if>

        <xsl:if match=".[@t='E']">
                <xsl:if match=".[vlr='1']">
                   <font face="Arial Narrow" color="#000000">LIGADO</font>
                </xsl:if>
                <xsl:if match=".[vlr='0']">
                   <font face="Arial Narrow" color="#000000">DESLIGADO</font>
                </xsl:if>
        </xsl:if>
        <xsl:if match=".[@t='A']">
                <xsl:if match=".[vlr='1']">
                   <font face="Arial Narrow" color="#000000">Auto</font>
                </xsl:if>
                <xsl:if match=".[vlr='0']">
             <font face="Arial Narrow" color="#000000">Manual</font>
                </xsl:if>
        </xsl:if>
</xsl:template>
</xsl:stylesheet>
```

A-4 Código fonte da página com gráfico Penchart

```
<HTML>
<table border="1" width="82%" cellspacing="0" cellpadding="2">
  <tr>
    <td width="50%">
<APPLET code="PenChart.class" archive=" webview.jar " width=500 height=350>
<PARAM name = "QtdValores" value = "10">
<PARAM name = "background" value = "Palha">
<PARAM name = "graphColor" value = "Azul">
<param name="IntervalX" value="0-10">
<param name="MaxEng" value="10">
<PARAM name = "axisColor" value = "Verde">
<PARAM name = "fontColor" value = "Vermelho">
<param name = "fontStyle" value = "10">
<param name="frequency" value="1">
<PARAM name = "fontSize" value = "20">
<PARAM name = "description" value = "Sistema de Controle de tensão">
<PARAM name = "descriptionX" value = "Tempo(s)">
<PARAM name = "descriptionY" value = "Tensao(V)">
<PARAM name = "DivX" value = "0">
<param name="DivY" value="10">
<PARAM name = "MinEng" value = "0">
<param name="Operand" value="M03">
<PARAM name = "Penmode" value = "true">
</APPLET>
    </td>
</table>
```

A-5 Cronograma

Tabela A.1 – Cronograma de desenvolvimento da monografia

	Assunto	Item do assunto	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ	JAN
1ª Meta	Pesquisa e estudo	Acompanhamento do funcionamento da planta									
2ª Meta	Implementação prática (ligação do WebGate, configuração e conexão com PC)	Alimentação do WebGate									
		WebGate - Configuração de número IP através de cabo serial									
		WebGate - Acesso via rede Ethernet									
	Estudo sobre as linguagens para implementação de aplicações integradas com o CLP	Estudo de linguagem HTML, XML, JAVA SCRIPT e LADDER									
	Estudo sobre funcionamento dos módulos, definição de variáveis. Implementação de rotinas em LADDER testando variaveis analógicas e digitais	CLP - Estudo do módulo PO3042									
		CLP - estudo do módulo PO2022									
		CLP - estudo do módulo PO1112									
		CLP - estudo do módulo PO2132									
		Implementação de rotianas em LADDER com variáveis digitais e analógicas									
	JAVA	JAVA - Envio de bibliotecas JAVA para o WebGate									
		JAVA - Testes de funcionamento com APPLET´s JAVA									
3ª Meta	FIREWALL	Estudo sobre tecnologias para implementação de FIREWALL´s									
		Implementação de FireWall									
	CLP e JAVA	Implementações práticas com CLP e JAVA em aplicações industriais									

A-6 Diagrama de ligações elétricas PO1112

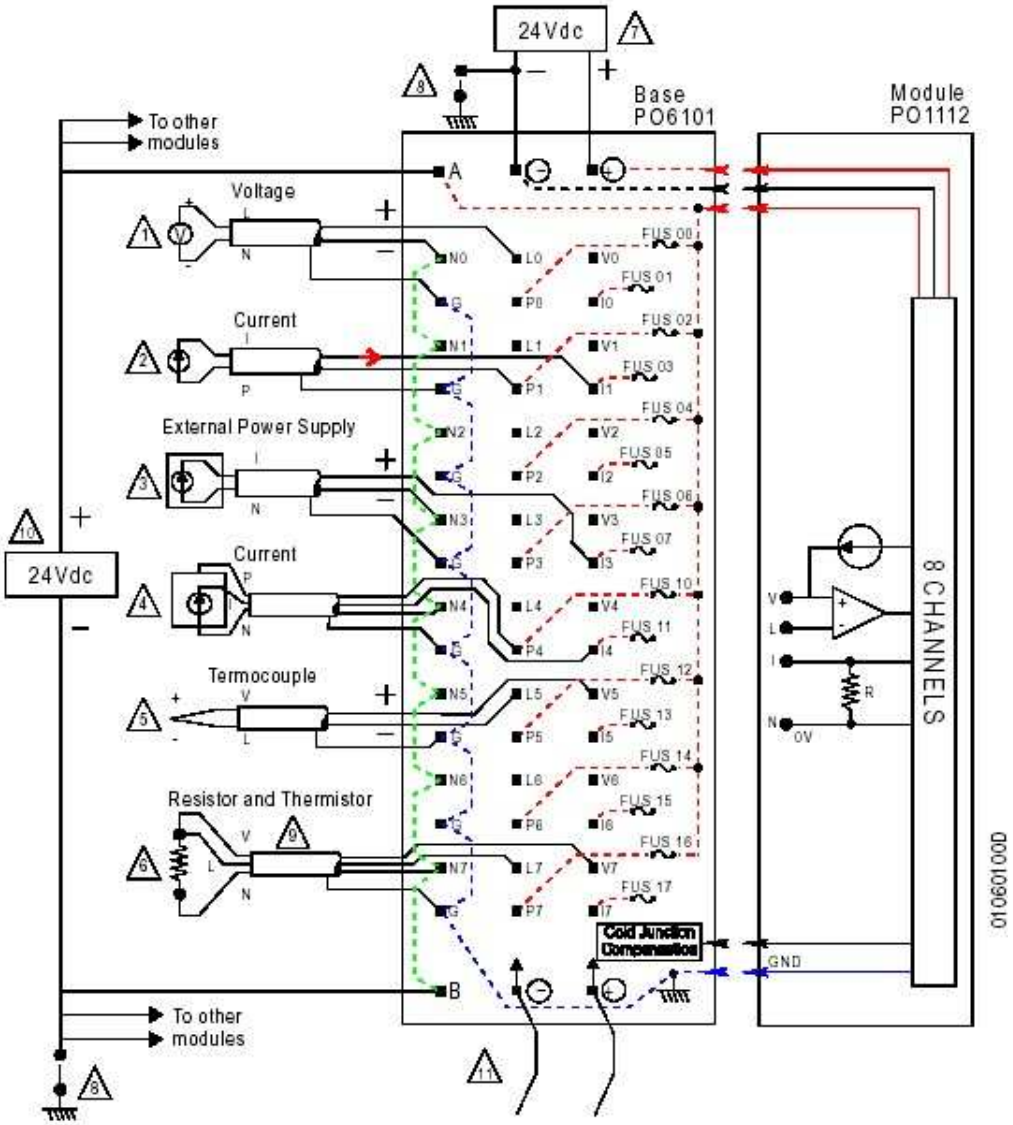


Figura A.8 – Diagrama de ligações elétricas do PO1112

A-7 Modos de tensão, corrente e resistência.

Tabela A.2 - Características do modo de tensão

	PO1112 – Modo Tensão		
Precisão	± 0,1 % do fundo de escala @ 25 °C ± 0,005% / °C do fundo de escala		
Resolução	16 bits Monotonicidade garantida sem códigos faltantes		
Impedância de entrada	1,1 MΩ		
Tensão máxima sem dano	± 30 V		
Filtragem	Constante de tempo configurável : 0,2 ms, 1,3 ms, 100 ms, 1 s ou 10 s		
Crosstalk DC a 100 Hz	- 30dB mim		
Tempo de atualização	25 ou 100 ms		
Escalas	Faixa	Contagem	Resolução
	-100 a +100 mV	-30.000 a 30.000	3,33 µV
	0 a 1 V	0 a 30.000	33,3 µV
	0 a 5 V	0 a 30.000	166,6 µV
	0 a 10 V	0 a 30.000	333 µV
	-1 a +1 V	-30.000 a 30.000	33,3 µV
	-5 a +5 V	-30.000 a 30.000	166,6 µV
	-10 a +10 V	-30.000 a 30.000	333 µV
Folga de escala	± 5%		
Indicação de Overange	se 5% (do fundo de escala) superior ou inferior a faixa de medição		

Tabela A.3 – Características do modo de corrente

	PO1112 – Modo Corrente		
Precisão	± 0,1 % do fundo de escala @ 25 °C ± 0,005% / °C do fundo de escala		
Resolução	16 bits Monotonicidade garantida sem códigos faltantes		
Impedância de entrada	50 Ω sem fusível (base PO6001 ou PO6051) 340 Ω com fusível 32 mA (base PO6101 ou PO6151)		
Corrente máxima contínua sem dano	40 mA		
Filtragem	Constante de tempo configurável : 0, 1.3 ms, 100 ms, 1 s ou 10 s		
Crosstalk @ 100hz	- 30dB mim		
Tempo de atualização	25 ou 100 ms		
Escalas	Faixa	Contagem	Resolução
	-1 a +1 mA	-30.000 a 30.000	0,033 µA
	0 a 20 mA	0 a 30.000	0,666 µA
	4 a 20 mA	0 a 30.000	0,533 µA
	-20 a +20 mA	-30.000 a 30.000	0,667 µA
Folga de escala	± 5%		
Indicação de Overange	se 5% (do fundo de escala) superior ou inferior a faixa de medição		
Detecção de cabo rompido	Indica o cabo de campo rompido (somente para a escala de 4 a 20 mA)		

Tabela A.4 – Características do mode de resistência

	PO1112 – Modo Resistência		
Precisão	± 0,1 % do fundo de escala @ 25 °C ± 0,006 % / °C do fundo de escala		
Resolução	16 bits Monotonicidade garantida sem códigos faltantes		
Impedância de entrada	1,1 MΩ		
Corrente de excitação	452 µA		
Técnica de medição	2 ou 3 fios		
Filtragem	Constante de tempo configurável : 1.3 ms, 100 ms, 1 s ou 10 s		
Crosstalk entre canais @ 100hz	- 30dB mim		
Detecção de resistência aberta	Indicado no diagnóstico se a resistência medida for 8% superior a faixa de medição		
Escalas	Faixa	Contagem	Resolução
	0 a 400 Ω	0 a 30.000	13,3 mΩ
	0 a 4000 Ω	0 a 30.000	133 mΩ
Tempo de atualização	100 ms		
Folga de escala	± 5%		
Cabo do sensor	3-fios, resistência máxima de cada fio = 20 Ω		
Indicação de Overange	se 5% (do fundo de escala) superior ou inferior a faixa de medição		

A-8 Problemas e soluções

Durante a ligação do CLP tivemos problemas no encaixe dos módulos com a base.

Tivemos que ativar o parâmetro habilita geral (P243) para fazer a velocidade do motor ser controlada pelo inversor.

WEBGATE

Não foi possível fazer a gravação de arquivos no *WebGate* utilizando os clientes FileZilla 3.1.2 *FTP* Cliente e *LeechFTP* 1.3 . Sempre ao tentar gravar é exibida uma mensagem dizendo que o serviço não está disponível.

Abaixo uma parte do log do programa FileZilla

Resposta: 421 Service not available, closing control connection.
Erro: Não pôde conectar no servidor
Status: Atrasando a conexão devido a tentativa de conexão que falhou anterioremente...
Status: Conectando com 192.168.0.181:21...

Entramos em contato com o Suporte da Altus (www.altus.com.br) através do e-mail altus@altus.com.br. Assim que o e-mail foi enviado eles responderam dizendo que o problema estava sendo analisado e que iriam entrar em contato. Um novo e-mail foi enviado informando mais detalhes do problema. Eles responderam que poderia ser o tamanho do arquivo porque a memória flash do WegGate tem capacidade máxima para 150Kbytes. Checamos o espaço disponível no *WebGate*, havia pouco mais de 50Kbytes, ou seja espaço suficiente já que estava tentando gravar um arquivo de 1Kbyte. Entramos

novamente em contato, agora através do suporte on-line (Chat) disponível na site da Altus. Conversamos com o analista que nos enviou por e-mail o programa WS_FTP. Testamos e a gravação funcionou. Com isso, concluímos que existe incompatibilidade entre o *WeGate* e outros clientes *FTP*.

CIMREX 12

Na programação da interface IHM foi necessário solicitar ao fabricante o DRIVER para atualização e o manual de configuração. Também tivemos problemas com o cabo de comunicação entre o CLP e a interface IHM Cimrex.

A interface IHM CIMREX 12 apresentou problemas com as teclas F2 e F3 que seriam usadas para SET e RESET. Entramos em contato com o suporte que nos orientou que poderia ser o programa ou algum problema físico. O programa não seria porque o mesmo programa está sendo usado na simulação da planta. Tentamos entrar no modo de diagnóstico pressionando SETA para direita e Seta para esquerda, mas a interface não responde. Caso ele tivesse respondido, deveria entrar no modo de teste e fazer uma verificação geral da interface. Pegamos a interface da planta para fazermos os testes na bancada. Utilizamos a interface para simulação, acionamento e o bloqueio de uma chave, com isso fizemos o monitoramento da variação da variável através do gráfico Penchart.

PO2132 – Modulo de saída analógica

Após toda conexão física e configuração utilizando o ProPronto (ALTUS), os módulos ficavam com o LED DG piscando o tempo todo. Tivemos que modificar no módulo PO2132 os canais 0,1,2 e 3 com valor **Corrente 4 a 20mA.**

PO1112 – Módulo de entrada analógica

Fizemos testes com resistor de 3,3kΩ e potenciômetro de 4k7. Configuramos os canais 0 e 1 para testar as resistências.

Problema: Ligamos o resistor e o potenciômetro na base, mas sempre o LED DG ficava diagnosticando aberto.

Solução: Depois de trocar alguns e-mails e tirar dúvidas online, o suporte da Altus informou que para resistência devemos fazer um curto ligando borne V ao L.

PO3045 - UNIDADE CENTRAL DE PROCESSAMENTO

Problema: Durante a transição para a UCP PO3045 tivemos um problema que sempre ao colocar no modo de execução o LED DG apresentava erro, cuja mensagem era : "CP SEM MÓDULO DE EXECUÇÃO E001".

Solução: Na tela principal de criação de projetos, devemos criar um módulo principal E001.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] Kurose, James F.; Ross Keth W. ***Redes de Computadores e a Internet: Uma abordagem top down;*** Tradução Arlete Simille Marques; revisão técnica Wagner Luiz Zucchi - 3ª Edição - São Paulo: Pearson Addison Wesley, 2006.

[2] Altus Sistemas de Informática S.A. **Manual de Utilização do *WebGate* – PO9900.** MU209690, 06/2003.

[3] Altus Sistemas de Informática S.A. **Manual de Utilização do *WebView* – MT7000**, MU299043, 07/2004

[4] Altus Sistemas de Informática S.A. **Manual de Utilização do PO3042 – UCPs Série PONTO MU209104**, 04/2005

[5] Manual for CIMREX-Terminais and CIMREX Prog.

[6] Altus Sistemas de Informática S.A. **Nota de aplicação NAP033: Desenvolvimento de Drivers de comunicação**,Revisão c, 10/04/2003

[7] Altus Sistemas de Informática S.A.**Nota de aplicação NAP080: Desenvolvimento de Páginas no PO9900 - *WEBGATE***, Revisão A, 05/03/2001.

[8] Altus Sistemas de Informática S.A. **Nota de aplicação NAP103: Configuração de Redes Ethernet**, Revisão A, 28/02/2007.

[9] Altus Sistemas de Informática S.A. **Manual de Utilização da Série Ponto**, Revisão C, 10/2001

[10] Stéfano, Eng. Antônio Geraldo. **Apostila sobre controladores programáveis (curso baseado no CP Altus)**, 05/2007.

[11] Souza, Luiz Edval. **Controladores Lógicos Programáveis**, 2001.

[12] Torres, Gabriel. **Redes de Computadores – Curso Completo**. Rio de Janeiro, Axcel Books, 2001

[13] Altus Sistemas de Informática S.A. **Características Técnicas PO1112 – CT 109312** Revisão G.

[14] Antonelli, Pedro Luis. **Introdução aos Controladores Lógicos Programáveis**, 1998.